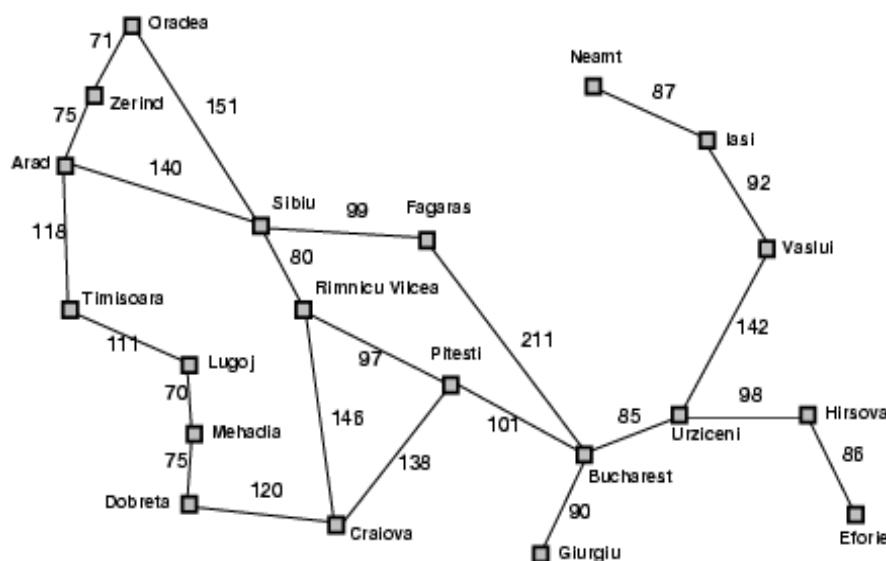


Berdoalah terlebih dahulu agar Anda sukses dalam ujian ini!

### A. Soal dari Masayu Leylia Khodra

1. Diberikan graf berikut ini:



| Straight-line distance to Bucharest |     |
|-------------------------------------|-----|
| Arad                                | 366 |
| Bucharest                           | 0   |
| Craiova                             | 160 |
| Dobreta                             | 242 |
| Eforie                              | 161 |
| Fagaras                             | 176 |
| Giurgiu                             | 77  |
| Hirsova                             | 151 |
| Lasi                                | 226 |
| Lugoj                               | 244 |
| Mehadia                             | 241 |
| Neamt                               | 234 |
| Oradea                              | 380 |
| Pitesti                             | 10  |
| Rimnicu Vilcea                      | 193 |
| Sibiu                               | 253 |
| Timisoara                           | 329 |
| Urziceni                            | 80  |
| Vaslui                              | 199 |
| Zerind                              | 374 |

Simpul pada graf menunjukkan kota, sedangkan bobot menunjukkan jarak dari suatu kota ke kota lainnya.

- a. Tentukanlah lintasan dari Arad ke Bucharest yang meminimumkan jarak tempuhnya dengan menggunakan *greedy*, *BFS*, *DFS*, *Branch & Bound* (B&B), dan Program Dinamis.

- Definisikanlah strategi yang digunakan untuk *greedy*, fungsi estimasi *cost* untuk B&B, dan tahapan & fungsi rekursif untuk Program Dinamis.

Hints: untuk Program Dinamis, sederhanakan dulu grafnya dan gunakan 3 tahapan.

- Khusus untuk *BFS*, *DFS*, dan B&B, gambarlah pohon ruang statusnya, dengan memberi nomor simpul berdasarkan urutannya relatif pada pohon tersebut (tidak mengacu ke penomoran pada pohon ruang status yang lengkap). Hanya simpul unik yang akan dicantumkan pada pohon tersebut. *State* menyatakan kota yang dicapai, sedangkan operator yang digunakan adalah *goto X* (dapat disingkat dengan huruf pertama kota yang dituju). Prioritas operator untuk level yang sama diberikan mulai dari kota di atas kota sekarang dengan arah lintasan jarum jam, misalnya operator untuk state Sibiu mulai dari F dulu, lalu R, A, dan O.

- Pencarian dihentikan saat lintasan pertama ke Bucharest didapatkan.

**(Skor: 5+5+5+15+15)**

- b. Jelaskan dengan disertai alasan strategi mana yang paling baik digunakan untuk menentukan lintasan terpendek ini. **(Skor: 5)**

2. Pencarian kalimat dapat dilakukan dengan *pattern* berupa *regular expression* (*regex*). Berikanlah pattern yang tepat dari bagian yang dihighlight pada teks berikut. **(Skor: 10)**

1. Informasi transaksi online di Indonesia didapatkan dari survei yang hanya melibatkan pembeli online dalam jumlah kecil (DailySocial, 2012).
2. Pengumpulan informasi transaksi dari para penjual online tidak efektif karena jumlah penjual yang banyak dan informasi tersebut merupakan data rahasia bagi para penjual.
3. Makalah ini mengeksplorasi analisis konten tweet dari twitter untuk mengumpulkan secara otomatis informasi transaksi online di Indonesia.
4. Pengguna twitter dari Indonesia telah mencapai 30 juta pada Juli 2012 (Semiocast, 2012).
5. Microblog ini telah digunakan para penggunanya untuk menulis berbagai aktifitas termasuk aktifitas dalam melakukan transaksi online.
6. Selain itu, transaksi online dilakukan pengguna twitter lebih banyak 24% dibandingkan rata-rata pengguna internet lainnya (Boorstin, 2012).
7. Analisis konten tweet berbahasa Indonesia sudah dilakukan untuk berbagai kepentingan seperti analisis opini dan sentimen (Romelta, 2012; Sunni & Widhyantoro, 2012; Aliandu, 2012), klasifikasi tweet kemacetan lalu lintas (Rodiyansyah, 2012), ekstraksi informasi kemacetan lalu lintas (Hasby & Khodra, 2013; Endarnoto dkk, 2011), ataupun peringkasan untuk menjelaskan trending topic pada twitter Indonesia (Winatmoko & Khodra, 2013).
8. Aplikasi analisis yang dikembangkan disebut dengan SaFE-F yang melakukan pencarian tweet (Search) dengan kata kunci tertentu, memfilter konten tweet yang relevan dengan kegiatan transaksi online (Filter), mengekstraksi informasi transaksi online (Ekstraksi), dan menyimpan hasil ekstraksi dalam bentuk yang lebih terstruktur (Filling).
9. Selain mengembangkan aplikasi SaFE-F, penelitian ini juga berkontribusi dalam mengkonstruksi korpus tweet yang telah dilabeli informasi transaksi online, dan mengembangkan model ekstraksi berbasis klasifikasi untuk mengekstraksi informasi transaksi online pada tweet berbahasa Indonesia (Hasby & Khodra, 2013).

## B. Soal dari Rinaldi Munir

3. Berikan satu buah kasus pencocokan *string* sedemikian sehingga kinerja algoritma KMP, Boyer-Moore, dan *Brute Force* sama! (yang dimaksud ‘sama’ adalah jumlah perbandingan karakter yang terjadi sama banyaknya). Tunjukkan dengan contoh sebuah *pattern* sepanjang enam karakter dan *text* sepanjang 10 karakter. Perlihatkan proses pencocokannya sehingga terlihat jumlah perbandingan dan pergeseran yang terjadi sama. (Skor: 10)
  4. Diberikan sebuah *text*: abacaabaccabacabaa dan *pattern*: abacab.
    - Hitung fungsi pinggiran
    - Gambarkan proses pencocokan *string* dengan algoritma KMP sampai *pattern* ditemukan
    - Berapa jumlah perbandingan karakter yang terjadi?(Skor: 5 + 10+5)
  5. Definisi persoalan *Integer Knapsack Decision Problem* adalah sebagai berikut: Apakah terdapat objek-objek (dengan masing-masing bobot  $w_i$  dan profit  $p_i$ ) yang dapat dimasukkan ke dalam sebuah *knapsack* sehingga total bobotnya tidak melebihi  $W$  tetapi total profitnya paling sedikit sebesar  $P$ . Jawaban terhadap persoalan keputusan ini adalah YA atau TIDAK.
- Integer Knapsack Decision Problem* termasuk ke dalam kelas NP. Algoritma di dalam kelas NP memiliki kompleksitas waktu polinom pada tahap verifikasinya, sedangkan pada tahap menerka adalah non-deterministik.

Tuliskan algoritma dalam bentuk fungsi (tidak usah terlalu rinci *pseudo-code* nya, bisa dalam bentuk kalimat deskriptif) pada tahap verifikasi tersebut yang mempunyai waktu polinom. *Header* fungsi sebagai berikut:

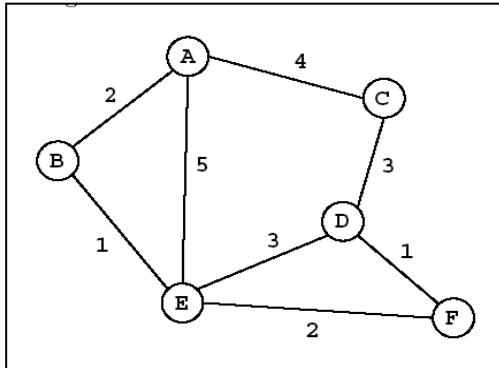
```
function verify( $W$ : integer;  $P$ : integer,  $S$ :string): boolean
{ Fungsi mengembalikan nilai true jika objek-objek yang direpresentasikan oleh string  $S$  memenuhi konstrain dan total profitnya paling sedikit sebesar  $P$ . Sebaliknya, fungsi mengembalikan nilai false. Konstrain pada persoalan knapsack adalah total bobot objek-objek yang dimasukkan ke dalam knapsack harus  $\leq W$ .
Contoh nilai string  $S$  yang dibangkitkan pada tahap menerka adalah 011101 yang artinya: 0 jika objek  $i$  dimasukkan, 1 jika objek  $i$  dimasukkan. Perhatikan bahwa  $S$  bisa sebuah non-sense string }
(Skor: 10)
```

6. Apa prediksi nilai Anda untuk kuliah ini? (A/AB/B/BC/C/D/E) (Skor: 2)

*Berdoalah terlebih dahulu agar Anda sukses dalam ujian ini!*

### **Bagian A (Soal dari Dr. Nur Ulfa Maulidevi)**

1. Dalam permainan *video game*, adakalanya entitas bergerak dalam *video game* perlu berpindah dari satu posisi ke posisi lain. Seringkali proses perpindahan perlu mengutamakan jalur terdekat atau biaya minimal karena berhubungan dengan poin yang diperoleh. Gambar di bawah ini menunjukkan contoh jalur yang mungkin dilewati oleh entitas bergerak dalam suatu *video game*. Suatu entitas akan berpindah dari posisi titik A menuju ke posisi titik F. Jika diperlukan informasi heuristik, nilai heuristik dari suatu simpul adalah **banyaknya busur minimal** yang menghubungkan titik tersebut ke titik tujuan.



Gambar 1. Graf lintasan pada suatu Video Game

Lengkapilah tabel berikut untuk mencari jalur dari simpul A ke simpul F dengan menggunakan beberapa pendekatan. Pencarian dihentikan ketika solusi pertama ditemukan.

| Iterasi | Uniform Cost Search  |   | Greedy Best First Search  |   | A Star  |   |
|---------|--|---|---|---|---|---|
|         | Simpul - Ekspan  | Simpul Hidup  | Simpul - Ekspan   | Simpul Hidup  | Simpul - Ekspan   | Simpul Hidup  |
|         | Formula: $f(n) = \dots$ {Isikan formula untuk mencari $f(n)$ dengan UCS} |   | Formula: $f(n) = \dots$ {Isikan formula untuk mencari $f(n)$ dengan Greedy Best First Search} |   | Formula: $f(n) = \dots$ {Isikan formula untuk mencari $f(n)$ dengan A Star} |   |
| 1       | A  | $B_A$<br>$f(B_A) = \dots$<br>$\dots$<br>$f(\dots) = \dots$<br>$\dots$<br>$f(\dots) = \dots$ | A   | $B_A$<br>$f(B_A) = \dots$<br>$\dots$<br>$f(\dots) = \dots$<br>$\dots$<br>$f(\dots) = \dots$ | A   | $B_A$<br>$f(B_A) = \dots$<br>$\dots$<br>$f(\dots) = \dots$<br>$\dots$<br>$f(\dots) = \dots$ |
| 2       | ...  | ...   |   |   |   |   |

|       |   |   |   |  |
|-------|---|---|---|--|
| ...   |   |   |   |  |
| Hasil | Jalur = A - ... - ...<br>Jarak = ...<br>Banyaknya iterasi hingga solusi pertama ditemukan = ... | Jalur = A - ... - ...<br>Jarak = ...<br>Banyaknya iterasi hingga solusi pertama ditemukan = ... | Jalur = A - ... - ...<br>Jarak = ...<br>Banyaknya iterasi hingga solusi pertama ditemukan = ... |  |

Catatan: Urutan anak adalah sesuai urutan abjad. Untuk tiap pendekatan, simpul yang sudah pernah di-ekspan dengan jalur yang sama tidak akan diperiksa lagi, dan tidak boleh membentuk sirkuit.

**(Nilai = 30)**

2. Dengan graf yang sama pada Gambar 1, selesaikan persoalan TSP yang dimulai dari simpul A dan berakhir di simpul A menggunakan pendekatan Branch and Bound. Fungsi pembatas yang digunakan adalah bobot minimum tur lengkap. Gambarkan pohon pencarian yang terbentuk, dan lengkapi tiap simpul dengan perhitungan untuk mendapatkan bobot pada tiap simpul tersebut. Pencarian dihentikan saat solusi pertama sudah ditemukan. Tuliskan jalur yang dihasilkan dan bobot dari jalur tersebut.

**(Nilai = 20)**

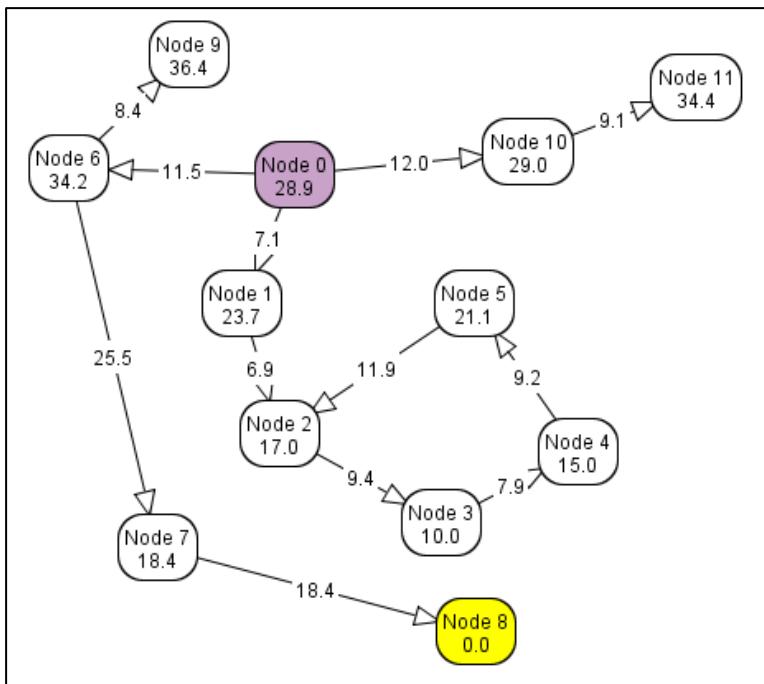
### **Bagian B (Soal dari Dr. Rinaldi Munir)**

3. Perhatikan kembali graf pada soal nomor 1. Tentukan tur terpendek yang dimulai dari simpul A, melalui simpul lainnya tepat satu kali, dan kembali lagi ke simpul A dengan total bobot minimal dengan metode Program Dinamis. **(Nilai = 20)**
4. Diberikan sebuah text: 000...0001 (sepanjang 1000 karakter) dan pattern 000000001 (sepanjang 10 karakter). Hitung jumlah perbandingan karakter yang terjadi jika pencocokan string dilakukan dengan:
  - Algoritma Brute Force
  - Algoritma Knuth-Morris-Pratt (hitung fungsi pinggiran lebih dahulu)
  - Algoritma Boyer-Moore (hitung fungsi *last occurance* lebih dahulu)**(Nilai = 15)**
5. (a) Gambarkan dalam diagram Venn hubungan antara kelas P, NP, dan NP-Complete  
 (b) Mengapa kelas persoalan P termasuk subset dari kelas NP?  
 (c) Sebutkan masing-masing 3 persoalan keputusan yang termasuk kelas P, kelas NP, dan kelas NPC, lengkap dengan deskripsi persoalannya. **(Nilai = 15)**
6. Apa perkiraan nilai anda untuk mata kuliah ini? (A/AB/B/BC/C/D/E) **(Nilai = 2)**

*Berdoalah terlebih dahulu agar Anda sukses dalam ujian ini!*

### **Bagian A (Soal dari Dr. Nur Ulfa Maulidevi)**

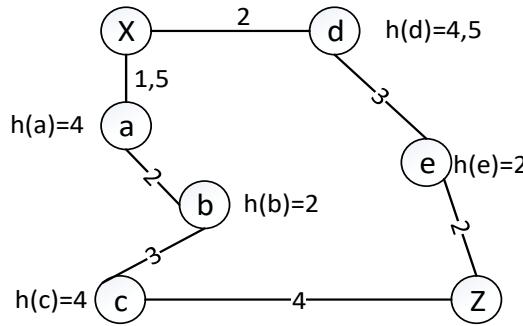
1. Pencarian jalur pada suatu graf dengan memanfaatkan algoritma A\* bertujuan untuk menghindari jalur yang ‘mahal’. Oleh karena itu nilai heuristik setiap simpul harus *admissible*. Dari Gambar 1 di bawah ini, tentukan apakah terdapat nilai heuristik yang tidak *admissible*, dan jelaskan dengan singkat. Nilai pada simpul adalah nilai heuristik, sedangkan nilai pada busur adalah biaya (cost) antar simpul. Pencarian jalur dimulai dari Node\_0 untuk menuju Node\_8.



Gambar 1. Graf berarah

**(Nilai = 10)**

2. Pada Gambar 2 berikut ini, terdapat graf yang menghubungkan simpul X ke simpul Z yang melalui beberapa simpul antara. Informasi mengenai jarak antar simpul dan nilai heuristik simpul (SLD dari simpul ke Z) sudah diberikan pada gambar tersebut.



Gambar 2. Graf lintasan X - Z

Lengkapilah tabel berikut untuk mencari jalur dari simpul X ke simpul Z dengan menggunakan beberapa pendekatan. Pencarian dihentikan ketika solusi pertama ditemukan.

| Iterasi | Greedy Best First Search  |   | A Star  |   |
|---------|---|---|---|---|
|         | Formula: $f(n) = \dots$ {Isikan formula untuk mencari $f(n)$ dengan Greedy Best First Search}   |   | Formula: $f(n) = \dots$ {Isikan formula untuk mencari $f(n)$ dengan A Star}                     |   |
|         | Simpul - Eksidan  | Simpul Hidup                                    | Simpul - Eksidan  | Simpul Hidup                                    |
| 1       | X   | $A_x$<br>$f(A_x) = \dots$<br>$f(\dots) = \dots$ | X   | $A_x$<br>$f(A_x) = \dots$<br>$f(\dots) = \dots$ |
| Dst...  |   |   |   |   |
| Hasil   | Jalur = X - ... - ...<br>Jarak = ...<br>Banyaknya iterasi hingga solusi pertama ditemukan = ... |   | Jalur = X - ... - ...<br>Jarak = ...<br>Banyaknya iterasi hingga solusi pertama ditemukan = ... |   |

Catatan: Urutan anak adalah sesuai urutan abjad. Untuk tiap pendekatan, simpul yang sudah pernah di-**ekspan** dengan jalur yang sama tidak akan diperiksa lagi, dan tidak boleh membentuk sirkuit.

(Nilai = 20)

3. Terdapat persoalan *Knapsack* dengan kapasitas total bobot *Knapsack* 10. Terdapat 4 barang yang tersedia dengan bobot dan profit seperti pada Tabel 1. Selesaikan persoalan Knapsack tersebut dengan pendekatan *Branch and Bound*, dengan fungsi pembatas :

total\_profit\_barang\_terpilih + (sisa\_bobot.yang\_masih\_memungkinkan \* nilai\_density\_terbaik\_berikutnya)

Tabel 1. Persoalan Knapsack

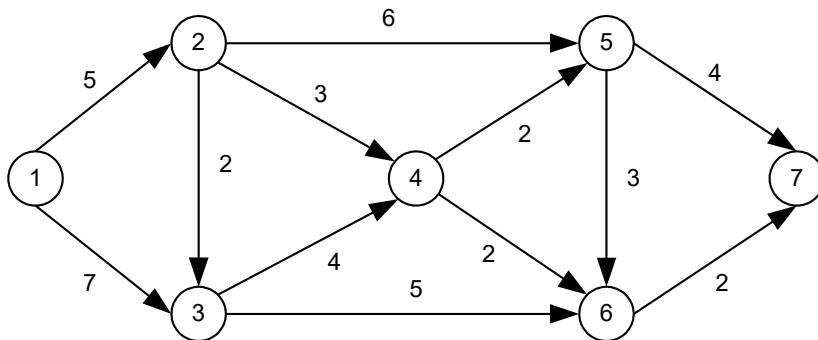
| Barang | Bobot | Profit | Profit/ Bobot<br>(Density) |
|--------|-------|--------|----------------------------|
| 1      | 4     | 40     | 10                         |
| 2      | 7     | 42     | 6                          |
| 3      | 5     | 25     | 5                          |
| 4      | 3     | 12     | 4                          |

Gambarkan pohon pencarian, lengkap dengan keterangan setiap simpul merepresentasikan apa dan perhitungan lengkap untuk setiap simpul yang dibangkitkan pada pohon pencarian.

(Nilai = 20)

## Bagian B (Soal dari Dr. Rinaldi Munir)

4. Carilah lintasan terpendek dari 1 ke 7 pada graf di bawah ini dengan algoritma Program Dinamis. Perlihatkan tabel perhitungannya (boleh pilih pendekatan maju atau mundur)



(Nilai = 20)

5. (a) Berikan contoh sebuah *pattern* yang panjangnya 6 karakter dan text yang panjangnya 10 karakter sedemikian sehingga pencocokan untuk kasus terburuk pada algoritma KMP sama dengan pencocokan untuk kasus terburuk algoritma *Brute-Force*. Hitung jumlah perbandingan karakter yang terjadi.  
(b) Diberikan sebuah *pattern*: EXAMPLE dan sebuah text: HERE IS A SIMPLE EXAMPLE. Jalankan algoritma Boyer-More untuk mencocokkan *pattern* dengan teks tersebut, perlihatkan proses pencocokannya *step by step*, lalu hitung jumlah perbandingan karakter yang terjadi.  
(Catatan: gambarkan *pattern* dan teks sebagai larik, lalu tuliskan angka 1, 2, 3, ... di atas *pattern* yang menyatakan jumlah perbandingan karakter yang terjadi)

(Nilai = 10 + 10)

6. Diberikan tiga soal pilihan berganda di bawah ini, pilihlah jawaban yang benar dan sertakan alasannya:

(6.1). Dengan mengasumsikan  $P \neq NP$ , maka dari jawaban berikut yang benar?

- (A) NP-complete = NP
- (B)  $NP\text{-complete} \cap P = \emptyset$
- (C)  $NP \cap NP\text{-complete} = P$
- (D)  $P = NP\text{-complete}$

(6.2) Manakah dari pernyataan berikut yang BENAR? (1) Persoalan menentukan adakah lintasan terpendek di dalam graf tak-berarah adalah persoalan P. (2) Persoalan menentukan apakah terdapat lintasan terpendek di dalam graf tak-berarah adalah persoalan NP. (3) Jika persoalan A adalah NP-Complete, maka terdapat algoritma non-deterministik dalam waktu polinom untuk memecahkan A.

- (A) 1, 2, dan 3
- (B) 1 dan 3
- (C) 2 dan 3
- (D) 1 dan 2

(6.3) Sebuah persoalan di dalam NP adalah NP-complete jika

- (A) ia dapat direduksi menjadi persoalan TSP dalam waktu polinom
- (B) Persoalan TSP dapat direduksi menjadi persoalan tersebut dalam waktu polinom
- (C) ia dapat direduksi menjadi sembarang persoalan lain di dalam NP dalam waktu polinom
- (D) Beberapa persoalan di dalam NP dapat direduksi menjadi persoalan tersebut dalam waktu polinom

(Nilai: 12)

7. Apa perkiraan nilai anda untuk mata kuliah ini? (A/AB/B/BC/C/D/E)

(Nilai = 2)

*Berdoalah terlebih dahulu agar Anda sukses dalam ujian ini!*

### **Bagian A (Path Finding dan Dynamic Programming)**

- (Nilai 20)** Terdapat persoalan 8-puzzle seperti pada Gambar 1. Gambar 1(a) adalah *start state* persoalan, dan gambar 1(b) adalah *goal state*. Ubin yang dapat bergerak adalah ubin kosong, dengan urutan pergerakan (jika diperlukan) adalah kiri, kanan, atas, bawah. Selesaikan persoalan 8-puzzle tersebut dengan teknik A\*. Biaya suatu *state* dari *start state* adalah banyaknya langkah ubin kosong yang telah dilakukan. Jika diperlukan heuristik, maka digunakan jumlah *manhattan distance* semua ubin **selain** ubin kosong. *Manhattan distance* sebuah ubin adalah banyaknya ubin secara horisontal dan vertikal terdekat untuk menuju posisi ubin yang sesuai dengan *goal state*.



Gambar 1. *Start State* dan *Goal State* Persoalan 8-Puzzle

Tuliskan langkah-langkah penyelesaiannya dengan menggambarkan pohon ruang pencarian, dan di setiap simpul pohon tunjukkan perhitungan nilai fungsi evaluasinya (yang menjadi penentu *state* berikutnya yang harus diperiksa). State yang sudah pernah dilalui tidak perlu diperiksa kembali. Setelah pencarian selesai, tuliskan langkah (*path*) pergeseran ubin kosong agar dari *start state* bisa mencapai *goal state*.

- (Nilai 20)** Terdapat barisan koin sebanyak n. Tiap koin bernilai sebuah bilangan bulat positif, dan nilai dari setiap koin tidak harus selalu berbeda. Tujuan dari persoalan ini adalah mendapatkan total koin terbesar, dengan batasan tidak boleh ada dua koin berurutan (bersebelahan) yang terambil sebagai solusi optimal. Berikut ini barisan tujuh koin yang dimulai dari indeks ke-0, dengan nilai koinnya. Indeks ke-0 artinya tidak ada koin yang terambil (koin dengan nilai nol).

|                          |   |   |   |   |   |    |   |   |
|--------------------------|---|---|---|---|---|----|---|---|
| Koin ke- atau Indeks ke- | 0 | 1 | 2 | 3 | 4 | 5  | 6 | 7 |
| Nilai koin ( $C_n$ )     |   | 5 | 1 | 2 | 9 | 10 | 9 | 2 |

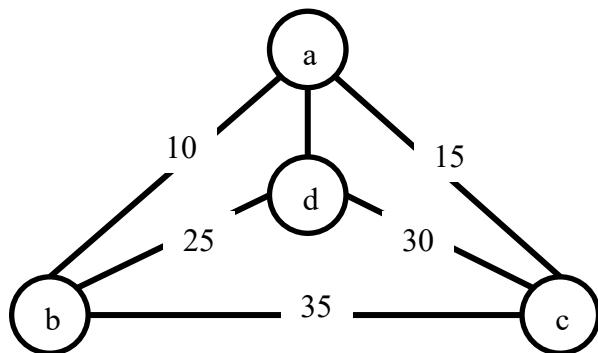
Persoalan tersebut akan diselesaikan dengan Program Dinamis.  $F(n)$  adalah jumlah maksimum dari koin yang bisa diambil pada barisan tersebut. Pendekatan yang bisa dilakukan adalah dengan membagi barisan koin tersebut menjadi dua buah kelompok, yaitu kelompok yang menyertakan koin terakhir ( $C_n+F(n-2)$ ), dan kelompok yang tidak menyertakan koin terakhir ( $F(n-1)$ ). Kita bisa memilih kelompok mana yang memberikan jumlah maksimum, pada setiap tahapnya.

- Tentukan struktur solusi optimal.
- Tentukan basis dan rekurens dari persoalan tersebut.
- Tuliskan perhitungan untuk mendapatkan nilai  $F(n)$  pada tiap tahap, sesuai dengan jawaban anda pada butir (b).
- Konstruksikan solusi optimal berdasarkan hasil perhitungan pada butir (c).

## Bagian B (Branch & Bound)

Untuk nomor 3 dan 4, jawaban ditulis dengan menggambarkan pohon ruang solusi dan menuliskan perhitungan cost setiap simpul. Jika ada simpul dengan cost sama pada level berbeda, pilihlah simpul dengan level yang lebih besar. Jika ada simpul dengan cost sama pada level yang sama, pilihlah path yang memilih simpul dengan abjad lebih kecil. Tanpa perhitungan detil, jawaban tidak akan dinilai.

- (Nilai 17.5)** Selesaikan persoalan TSP berikut dengan Branch & Bound dengan reduced cost matrix atau bobot tur lengkap. Simpul awal adalah a. Sebelum menjawab, tuliskan terlebih dahulu fungsi pembatas yang akan digunakan untuk menentukan cost simpul. Berikanlah semua tour terpendek yang dihasilkan.



- (Nilai 17.5)** Selesaikan persoalan Knapsack berkapasitas 100 dengan Branch & Bound untuk mendapatkan semua solusi optimal. Gunakanlah fungsi pembatas yang dipelajari di kuliah yaitu  $c(i)=f(i)+g(i)$  dengan  $f(i)$  adalah total nilai yang didapatkan sampai dengan simpul i, sedangkan  $g(i)$  adalah selisih kapasitas dengan total berat sampai simpul i dikali dengan rasio nilai dan berat barang berikutnya. Perhatikan syarat yang perlu dipenuhi untuk fungsi pembatas ini. Berikanlah semua solusi optimalnya dalam format  $(x_1, x_2, x_3, x_4, x_5, x_6)$

| Objek ke- | berat | nilai |
|-----------|-------|-------|
| 1         | 100   | 40    |
| 2         | 50    | 35    |
| 3         | 45    | 18    |
| 4         | 20    | 4     |
| 5         | 10    | 10    |
| 6         | 5     | 2     |

## **Bagian C (Pattern Matching dan NP)**

1. **(Nilai 15)** Diberikan teks “a long text with words.” dan pola “with”. Lakukan pencocokan pattern dengan text sampai ketemu dengan algoritma Brute Force, KMP, dan Boyer-Moore. Hitung jumlah perbandingan karakter yang terjadi. Dalam menjawab soal ini, hitung fungsi pinggiran dan fungsi last occurrence.

Teks:

|   |  |   |  |   |  |   |  |   |  |   |  |   |  |   |  |   |  |   |  |   |  |   |  |   |  |   |  |   |  |   |  |   |  |   |  |   |
|---|--|---|--|---|--|---|--|---|--|---|--|---|--|---|--|---|--|---|--|---|--|---|--|---|--|---|--|---|--|---|--|---|--|---|--|---|
| a |  | i |  | o |  | n |  | g |  | t |  | e |  | x |  | t |  | w |  | i |  | t |  | h |  | w |  | o |  | r |  | d |  | s |  | . |
|---|--|---|--|---|--|---|--|---|--|---|--|---|--|---|--|---|--|---|--|---|--|---|--|---|--|---|--|---|--|---|--|---|--|---|--|---|

Pola:

|   |  |   |  |   |  |   |
|---|--|---|--|---|--|---|
| w |  | i |  | t |  | h |
|---|--|---|--|---|--|---|

2. **(Nilai 10)** Misalkan persoalan *A* adalah **NP-complete** dan persoalan *B* adalah **NP** tapi tidak perlu **NP-complete**. Manakah dari pernyataan-pernyataan berikut yang benar? Jawaban bisa lebih dari satu. Jelaskan alasannya.  
I. Algoritma dengan waktu polinom untuk *A* mengimplikasikan bahwa  $P = NP$ .  
II. Algoritma dengan waktu polinom untuk *B* mengimplikasikan bahwa  $P = NP$ .  
III. Algoritma dengan waktu polinom untuk *A* mengimplikasikan algoritma dengan waktu polinom untuk *B*.

3. Apa perkiraan nilai anda untuk mata kuliah ini? (A/AB/B/BC/C/D/E)

**(Nilai = 2)**

Ujian Akhir Semester IF2211 Strategi Algoritma

Senin, 14 Mei 2018

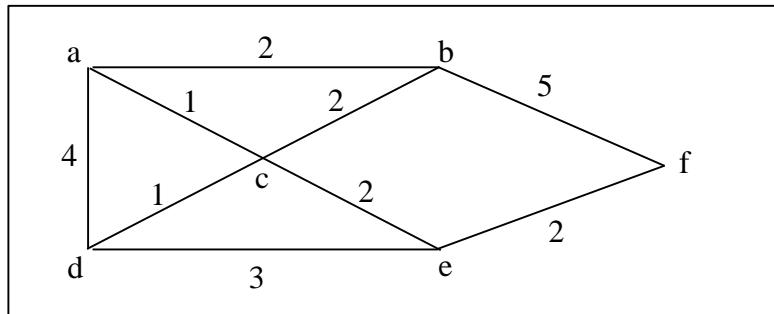
Waktu: 150 menit

Dosen: Masayu Leylia Khodra, Nur Ulfa Maulidevi, Rinaldi Munir

*Berdoalah terlebih dahulu agar Anda sukses dalam ujian ini!*

**Bagian A (Backtracking dan Route Planning)**

Terdapat sebuah graf sebagai berikut.



Gambar 1. Graf Tak Berarah

- (Nilai 15)** Tentukan ‘Hamiltonian Circuit’ dari graf pada Gambar 1, yang diawali pada simpul **a** dan berakhir di simpul **a** dengan menggunakan pendekatan Backtracking. Tentukan:
  - Representasi Solusi Persoalan (Nilai 3);
  - Deskripsi Fungsi Pembatas (Nilai 2);
  - Pohon ruang pencarian lengkap dengan nomor simpul yang menunjukkan urutan pembangkitan. (Nilai 10) Pencarian dihentikan ketika sudah mendapatkan sebuah solusi.

Catatan: prioritas simpul tetangga (atau simpul anak) yang dibangkitkan sesuai urutan abjad, dan ingat perbedaan antara ‘Hamiltonian Circuit’ dan ‘Traveling Salesperson Problem’

- (Nilai 15)** Gunakan pendekatan **UCS**, **Greedy Best First Search**, dan **A\*** untuk menentukan jalur optimal dari simpul **a** menuju simpul **f**, dengan melengkapi tabel di bawah ini.

| Iterasi | Uniform Cost Search (UCS)  |   | Greedy Best First Search  |   | A *   |   |
|---------|--|---|---|---|---|---|
|         | Formula: $f(n) = \dots$ {Isikan formula untuk mencari $f(n)$ dengan UCS} |   | Formula: $f(n) = \dots$ {Isikan formula untuk mencari $f(n)$ dengan Greedy Best First Search} |   | Formula: $f(n) = \dots$ {Isikan formula untuk mencari $f(n)$ dengan A Star} |   |
|         | Simpul - Ekspan  | Simpul-Hidup  | Simpul-Ekspan   | Simpul-Hidup  | Simpul - Ekspan   | SimpulHidup   |
| 1       | a  | $b_a$<br>$f(b_a) = \dots$<br>$c_a$<br>$f(c_a) = \dots$<br>$d_a$<br>$f(d_a) = \dots$ | a   | $b_a$<br>$f(b_a) = \dots$<br>$c_a$<br>$f(c_a) = \dots$<br>$d_a$<br>$f(d_a) = \dots$ | a   | $b_a$<br>$f(b_a) = \dots$<br>$c_a$<br>$f(c_a) = \dots$<br>$d_a$<br>$f(d_a) = \dots$ |
| 2       | ...  | ...   |   |   |   |   |
| ...     |  |   |   |   |   |   |
| Hasil   | Jalur = a - ... - ...<br>Jarak = ...                                     | Jalur = a - ... - ...<br>Jarak = ...  | Jalur = a - ... - ...<br>Jarak = ...  |   |   |   |

Catatan: Jika diperlukan heuristik, nilai heuristik sebuah simpul adalah banyaknya busur minimal yang menghubungkan simpul tersebut ke simpul tujuan. Pencarian dihentikan ketika

sudah mencapai simpul tujuan. Jika terdapat nilai  $f(n)$  terkecil yang sama, maka prioritas simpul yang diperiksa sesuai urutan abjad. Simpul yang sudah diekspan tidak perlu dimasukkan dalam agenda (daftar simpul-hidup).

### **Bagian B (Branch&Bound dan Dynamic Programming)**

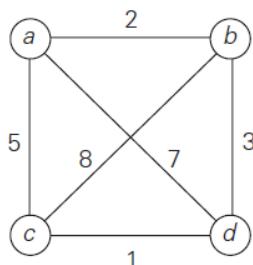
Untuk Program Dinamis (PD), berikanlah terlebih dahulu fungsi rekurensnya dan bentuklah solusinya per tahap. Solusi tanpa tahapan dan fungsi rekurensnya tidak akan diperiksa.

Untuk Branch and Bound (B&B), definisikanlah fungsi pembatas dan nilai setiap simpulnya sesuai fungsi objektifnya, lalu bentuklah pohon ruang status dinamis dengan nomor simpul menyatakan urutan pembangkitan. Nilai untuk akar tetap dihitung. Solusi tanpa kedua definisi tersebut tidak akan diperiksa.

3. **(Nilai 20)** Selesaikanlah persoalan Integer Knapsack berikut untuk memaksimumkan profit yang didapatkan dengan Program Dinamis (PD) **dan** Branch and Bound (B&B). Kapasitas maksimum adalah 3.

| Objek | Bobot | Profit |
|-------|-------|--------|
| A     | 1     | 5      |
| B     | 2     | 14     |
| C     | 1     | 6      |

4. **(Nilai 15)** Selesaikanlah persoalan TSP berikut untuk meminimumkan jarak sirkuit hamilton yang dilalui dengan Program Dinamis (PD) **atau** Branch and Bound (B&B). Simpul awal = a.



### **Bagian C (Pattern Matching dan Teori P dan NP)**

5. **(Nilai 25)**

- (a) Sebuah string biner sepanjang 100 bit semuanya terdiri dari bit 0. Hitung jumlah perbandingan karakter pada pencocokan *pattern* dengan algoritma *Brute Force* jika *pattern* adalah (i) 0001 dan (ii) 1000.
- (b) Pertanyaan yang sama dengan (a) jika algoritma yang digunakan adalah Boyer-Moore
- (c) Diberikan teks “HERE IS A SIMPLE EXAMPLE” dan pola “EXAMPLE”. Gambarkan proses pencocokan pola dengan teks sampai ketemu dengan algoritma KMP dan Boyer-Moore. Hitung jumlah perbandingan karakter yang terjadi. Dalam menjawab soal ini, hitung fungsi pinggiran dan fungsi *last occurrence*.

Teks:

|   |   |   |   |  |   |   |  |   |  |   |   |   |   |   |   |  |   |   |   |   |   |   |   |
|---|---|---|---|--|---|---|--|---|--|---|---|---|---|---|---|--|---|---|---|---|---|---|---|
| H | E | R | E |  | I | S |  | A |  | S | I | M | P | L | E |  | E | X | A | M | P | L | E |
|---|---|---|---|--|---|---|--|---|--|---|---|---|---|---|---|--|---|---|---|---|---|---|---|

Pola:

|   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|
| E | X | A | M | P | L | E |
|---|---|---|---|---|---|---|

6. **(Nilai 10)** Diberikan beberapa buah pernyataan di bawah ini tentang  $P$ ,  $NP$ , dan  $NP$ -complete. Tentukan pernyataan mana saja yang benar (sebutkan nomornya saja).
- (i)  $P$  Problem adalah himpunan semua persoalan apapun dengan kompleksitas waktu polinomial
  - (ii)  $NP$  adalah singkatan dari Non-Polynomial
  - (iii) Persoalan di dalam kelas  $NP$  memiliki waktu polinomial pada tahap verifikasi suatu solusi.
  - (iv) Sebuah persoalan  $X$  dikatakan  $NP$ -complete jika  $X$  termasuk ke dalam kelas  $NP$  dan  $X$  dapat direduksi menjadi persoalan di dalam  $NP$  lainnya dalam waktu polinomial.
  - (v) Jika  $A$  adalah sebuah persoalan di dalam  $NP$ -complete dan  $B$  adalah persoalan  $NP$  tapi tidak perlu  $NP$ -complete, maka jika  $A$  dapat diselesaikan dalam waktu polinomial maka implikasinya  $P = NP$ .
7. **(Nilai 2)** Apa perkiraan nilai anda untuk mata kuliah ini? (A/AB/B/BC/C/D/E)

Ujian Akhir Semester IF2211 Strategi Algoritma

Kamis, 9 Mei 2019

Waktu: 150 menit

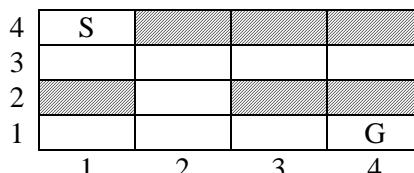
Dosen: Masayu Leylia Khodra, Nur Ulfa Maulidevi, Rinaldi Munir

*Berdoalah terlebih dahulu agar Anda sukses dalam ujian ini!*

**Bagian A (Backtracking, UCS, Greedy Best First, dan A\*)**

1. Terdapat sebuah labirin sederhana seperti pada gambar 1. Titik S (Start) berada pada posisi (1,4), dan titik G (Goal) berada pada posisi (4,1). Sel yang diarsir adalah sel yang tidak bisa dilewati. Persoalan yang akan diselesaikan adalah menemukan jalur dari S menuju G dengan menggunakan beberapa teknik pencarian. Jarak dari satu titik ke titik berikutnya adalah 1 (satu) satuan jarak. Jika diperlukan heuristik suatu titik  $(x',y')$ , digunakan jarak *Manhattan Distance*, dengan formula sebagai berikut.

$$h(x',y') = (\text{selisih } x' \text{ dengan posisi } x \text{ titik Goal}) + (\text{selisih } y' \text{ dengan posisi } y \text{ titik Goal})$$



Gambar 1. Labirin Sederhana

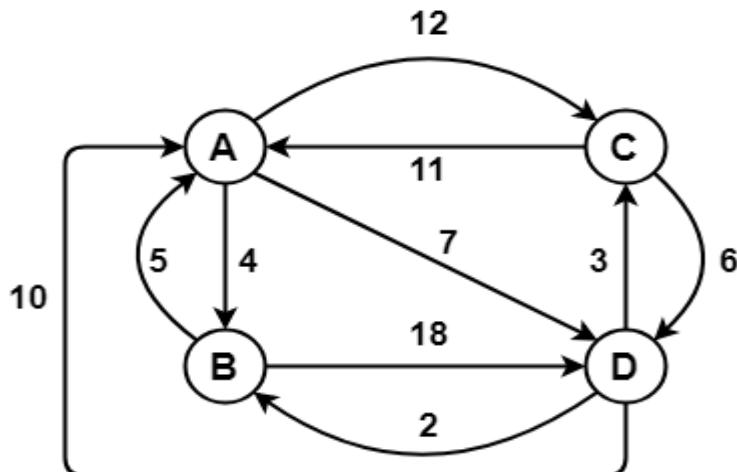
Operasi yang bisa dilakukan adalah bergerak *east* (posisi x bertambah 1), *south* (posisi y berkurang 1), *west* (posisi x berkurang 1), dan *north* (posisi y bertambah 1). Jika diperlukan, **urutan prioritas operasi** yang dilakukan adalah *east, south, west, north*.

- a. **(Nilai 7)** Buatlah pohon pencarian jalur ke titik Goal (4,1) dengan menggunakan teknik **Backtracking**, dimulai dari titik (1,4). Tulislah nomor urutan pembangkitan pada setiap simpul pohon pencarian. Pencarian dihentikan ketika sudah mencapai titik G. Kemudian tuliskan hasil **urutan aksi** yang dilakukan untuk mencapai G dari S.
- b. **(Nilai 7)** Tentukan nilai heuristik dari setiap titik yang bisa dilewati pada gambar 1, dan sebutkan titik-titik dengan nilai heuristik yang tidak admissible jika ada.
- c. **(Nilai 21)** Lengkapilah tabel berikut untuk mencari jalur dari titik S(1,4) ke titik G(4,1) dengan menggunakan beberapa pendekatan. Pencarian dihentikan ketika solusi pertama ditemukan.

| Iterasi | Uniform Cost Search  |  | Greedy Best First Search  |  | A Star  |  |
|---------|--|--|---|--|---|--|
|         | Formula: $f(n) = \dots$ {Isikan formula untuk mencari $f(n)$ dengan UCS} |  | Formula: $f(n) = \dots$ {Isikan formula untuk mencari $f(n)$ dengan Greedy Best First Search} |  | Formula: $f(n) = \dots$ {Isikan formula untuk mencari $f(n)$ dengan A Star} |  |
|         | Titik - Eksplan  | Titik Hidup (tuliskan semua titik hidup dan nilai $f(n)$ untuk tiap titik hidup) | Titik - Eksplan   | Titik Hidup (tuliskan semua titik hidup dan nilai $f(n)$ untuk tiap titik hidup) | Titik - Eksplan   | Titik Hidup (tuliskan semua titik hidup dan nilai $f(n)$ untuk tiap titik hidup) |
| 1       | (1,4)  |  | (1,4)   |  | (1,4)   |  |
| 2       | ...  | ...  |   |  |   |  |
| ...     |  |  |   |  |   |  |
| Hasil   | Jalur = (1,4) - ... - ...<br>Jarak = ...                                 |  | Jalur = (1,4) - ... - ...<br>Jarak = ...  |  | Jalur = (1,4) - ... - ...<br>Jarak = ...                                    |  |

## Bagian B (Branch&Bound dan Dynamic Programming)

2. Persoalan TSP berikut meminimumkan jarak sirkuit hamilton, dengan simpul awal = A.



- (a) **(Nilai 2.5)** Dengan menggunakan metode reduced cost matrix, berikanlah proses perhitungan taksiran cost untuk simpul akar.
- (b) **(Nilai 10)** Selesaikanlah persoalan TSP tersebut dengan metode bobot tur lengkap. Bentuklah pohon ruang status dinamis dengan nomor simpul menyatakan urutan pembangkitan. Solusi tanpa perhitungan cost per simpul tidak akan diperiksa.
- (c) **(Nilai 10)** Selesaikanlah persoalan TSP tersebut dengan Program Dinamis (PD). Solusi tanpa tahapan dan fungsi rekurensnya tidak akan diperiksa.
3. Untuk menyelesaikan persoalan 15-puzzle berikut ini dengan Branch and Bound (B&B), jawablah pertanyaan berikut ini:

- (a) **(Nilai 2.5)** Berikanlah definisi taksiran cost untuk simpul yang dibangkitkan.
- (b) **(Nilai 10)** Dengan asumsi puzzle di sebelah kiri dapat mencapai goal state (gambar kanan), bentuklah pohon ruang status dinamis dengan nomor simpul menyatakan urutan pembangkitan. Solusi tanpa perhitungan cost per simpul tidak akan diperiksa.



## Bagian C (Pattern Matching dan Teori P dan NP)

4. **(Nilai 3 + 15 + 5)**
- (a) Diberikan  $P = 00001$  dan  $T = 0000000000000001$ . Misalkan digunakan algoritma *Brute Force* untuk pencocokan string. Berapa jumlah perbandingan karakter yang terjadi?
- (b) Diberikan  $P = 10010001$  dan  $T = 100100100100010111$ . Gambarkan/perlihatkan proses pencocokan string  $P$  pada teks  $T$  masing-masing dengan algoritma *Brute Force*, KMP, dan Boyer- Moore. Gunakan angka-angka 1, 2, 3, ... untuk memperlihatkan jumlah perbandingan (seperti slide kuliah). Berapa jumlah perbandingan karakter yang terjadi?
- (c) Tulislah notasi *regex* untuk mengenali:

- Sembarang IP address (misalnya 012.345.678.912)
  - Sembarang alamat email (misalnya sabiyan.manis@itb.ac.id)
5. **(Nilai 7)** Diberikan beberapa buah pernyataan di bawah ini tentang  $P$ ,  $NP$ , dan  $NP$ -complete. Tentukan pernyataan mana saja yang benar dan mana yang salah. Jawab sbb: (i) B, (ii) S, dst.
- (i)  $P$  adalah himpunan semua persoalan apapun yang memiliki kebutuhan waktu dalam polinomial
  - (ii)  $NP$  adalah himpunan semua persoalan keputusan yang memiliki kebutuhan waktu non-polinomial
  - (iii) *Halting Problem* tidak termasuk ke dalam kelas  $NP$
  - (iv) Algoritma non-deterministik selalu memiliki tahap verifikasi dalam waktu polinomial.
  - (v) Sebuah persoalan  $X$  dikatakan  $NP$ -complete jika  $X$  termasuk ke dalam kelas  $NP$  dan beberapa persoalan di dalam  $NP$  lainnya dapat direduksi menjadi instans persoalan  $X$  dalam waktu polinomial.
  - (vi) Jika  $A$  adalah sebuah persoalan di dalam  $NP$ -complete dan  $B$  adalah persoalan  $NP$  tapi tidak perlu  $NP$ -complete, maka jika  $B$  dapat diselesaikan dalam waktu polinomial maka  $A$  juga dapat diselesaikan dalam waktu polinomial.
  - (vii)  $P = NP$  jika dan hanya jika persoalan di dalam  $NP$ -complete dapat diselesaikan dalam waktu polinomial.
6. **(Nilai 2)** Apa perkiraan nilai anda untuk mata kuliah ini? (A/AB/B/BC/C/D/E)

# Ujian Akhir Semester IF2211 Strategi Algoritma Semester 2 2020-2021

UAS IF2211 dilaksanakan dengan waktu 120 menit (13.00 - 15.00 WIB). Ujian bersifat individu dan tutup buku.

Terdapat tiga bagian dalam ujian ini. Bagian pertama adalah identitas dan pernyataan kejujuran, bagian kedua adalah soal isian (essay), dan bagian ketiga adalah soal pilihan. Pastikan anda 'submit' hasil pekerjaan setelah menyelesaikan UAS.

Email responden (**null**) dicatat saat formulir ini dikirimkan.

\* Wajib

1. Email \*

---

2. Nama \*

---

3. NIM \*

---

4. Kelas \*

*Tandai satu oval saja.*

- K1
- K2
- K3
- K4

5. Tulis ulang pernyataan berikut: "Saya menyatakan bahwa saya mengerjakan UAS ini dengan sejujur-jujurnya, tanpa bantuan orang lain dan tanpa menggunakan cara yang tidak dibenarkan. Apabila di kemudian hari diketahui saya mengerjakan UAS ini dengan cara yang tidak jujur, saya bersedia mendapatkan konsekuensinya, yaitu mendapatkan nilai E pada mata kuliah IF2211 Semester 2 2020/2021. " \*

---

---

---

---

---

#### Soal Essay

6. Pencocokan string sangat penting aplikasinya dalam bidang Sistem Temu Balik Informasi, misalkan internet search engine. Berdasarkan apa yang sudah kalian pelajari terkait pencocokan string, a) usulkan bagaimana struktur penyimpanan teks/string dari keyword-keyword yang ada di seluruh koleksi dokumen dalam suatu search-engine sehingga waktu yang diperlukan untuk melakukan pencocokan antara keyword di query dan keyword di koleksi dokumen bisa optimal dari segi waktu. b) Berikan contoh struktur penyimpanan keyword/string tersebut untuk beberapa keyword sehingga akan memperjelas jawaban anda di bagian a tersebut. c) berikan contoh jalannya pencarian/pencocokan keyword di query dan di koleksi dokumen tersebut.

---

---

---

---

---

7. (Sum-of-subset problem) Diberikan tiga buah bobot yaitu  $w_1=2$ ,  $w_2=4$ ,  $w_3=5$ . Kita akan mencari himpunan bagian bobot yang jumlahnya sama dengan  $m = 9$ . Kita akan menyelesaikan persoalan ini menggunakan algoritma backtracking dengan fungsi pembatas (bounding function) seperti yang sudah dijelaskan di dalam materi kuliah. Setelah dibangun pohon ruang statusnya, tuliskan (i) jumlah simpul di dalam pohon ruang status lengkap, (ii) berapa banyak simpul yang dibunuh (yang ditandai dengan B), (iii) berapa banyak simpul yang tidak dibunuh setelah semua solusi ditemukan, (iv) vektor solusi dalam bentuk  $X = (x_1, x_2, x_3)$ ,  $x_i$  bernilai 0 atau 1. Jawaban untuk soal ini berupa jawaban singkat dengan format  $[N, N1, N2, (x_1, x_2, x_3)]$ . Contoh jawaban adalah  $[12, 4, 6, (0, 1, 0)]$

$[15, 4, 5, (0, 1, 1)]$

## 8. Koefisien Binomial

Koefisien binomial diberikan oleh  $\binom{n}{k} = \frac{n!}{k!(n-k)!}$  untuk  $0 \leq k \leq n$ , dan dapat diformulasikan secara rekursif sebagai berikut:

$$\binom{n}{k} = \begin{cases} \binom{n-1}{k-1} + \binom{n-1}{k} & , 0 < k < n \\ 1 & , k = 0 \text{ atau } k = n \end{cases}$$

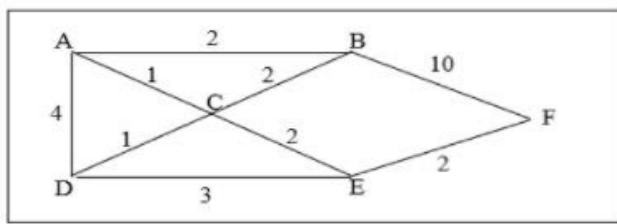
sehingga koefisien binomial ini bisa diselesaikan dengan algoritma Dynamic Programming sebagai berikut:

```
function bin2(n, k: integer): integer;
var
  i,j: index;
  B: array[0..n,0..k] of integer;
begin
  for i:= 0 to n do
    for j:= 0 to minimum(i,k) do
      if j = 0 or j = i then
        B[i,j]:= 1
      else
        B[i,j]:= B[i-1,j-1] + B[i-1,j]
      end
    end;
  bin2:= B[n,k]
end;
```

Contoh:  
 $B[7, 6] = B[6, 5] + B[6, 6] = ?$

- Dengan algoritma di atas, hitung  $B[3, 2] = \binom{3}{2}$  dengan menuliskan setiap  $B[i, j]$  untuk  $0 \leq i \leq n$  dan  $0 \leq j \leq k$  seperti pada contoh (memperlihatkan dekomposisi jika ada dan nilai akhirnya).
  - Karena memiliki bentuk rekursif, koefisien dapat juga diselesaikan dengan pendekatan Divide & Conquer. Mana algoritma perhitungan koefisien binomial yang lebih efisien, apakah dengan pendekatan Dynamic Programming atau Divide & Conquer, dan jelaskan mengapa bisa lebih efisien?
- 
- 
- 
-

9. Terdapat sebuah graf tidak berarah seperti pada Gambar 1 berikut ini. Simpul merepresentasikan kota, dan bilangan yang terdapat pada sisi adalah jarak antara dua kota. Persoalan yang ingin diselesaikan adalah mencari jalur dari kota A ke kota F, dan diharapkan dapat memberikan jalur dengan jarak minimum. Pendekatan untuk mencari jalur dengan jarak minimum adalah A star ( $A^*$ ). Jika diperlukan, nilai heuristik sebuah simpul adalah banyaknya sisi (edge) minimal dari simpul tersebut ke simpul tujuan. Tuliskan proses pencarian jalur dari kota A ke kota F seperti pada gambar. Catatan: urutan prioritas simpul sesuai dengan urutan abjad jika terdapat lebih dari satu simpul yang memiliki  $f(n)$  minimum.



Gambar 1.

A\*

Iterasi 1:

Simpul Ekspan (Simpul yang diperiksa) = A

Nilai  $f(n)$  untuk simpul hidup $f(B[A]) = \dots$  $f(C[A]) = \dots$  $f(D[A]) = \dots$ 

Iterasi 2:

Simpul Ekspan (Simpul yang diperiksa) = [tuliskan simpul yang diperiksa berikutnya]

Nilai  $f(n)$  untuk simpul hidup (tuliskan nilai  $f(n)$  untuk semua simpul hidup)

...

Iterasi 3:

Dst.... (Catatan: pencarian dihentikan ketika simpul ekspan/ yang diperiksa = F)

Jalur hasil pencarian dengan  $A^* = \dots$ Jarak jalur hasil pencarian dengan  $A^* = \dots$ 

Soal Pilihan

10. Backtracking adalah strategi algoritma yang banyak dipakai di bidang Artificial Intelligence. Pilihlah satu jawaban yang PALING TEPAT di bawah ini :

*Tandai satu oval saja.*

- Kemampuan strategi algoritma backtracking yang cara kerjanya menyerupai cara kerja otak manusia.
- Kemampuan strategi algoritma backtracking dalam menyelesaikan banyak masalah
- Kemampuan strategi algoritma backtracking dalam menyelesaikan masalah-masalah terkait numerik
- Kemampuan strategi algoritma backtracking dalam menyelesaikan masalah-masalah terkait symbolic
- Kemampuan strategi algoritma backtracking dalam menyelesaikan masalah-masalah yang kombinatorial.
- Bahasa pemrograman khusus AI yaitu Prolog menggunakan konsep backtracking.

11. Manakah di bawah ini persoalan yang hasilnya paling jelek jika diterapkan strategi algoritma backtracking ?

*Tandai satu oval saja.*

- Persoalan permainan Sudoku, mengisikan angka-angka sehingga jumlah angka tersebut dalam beberapa arah adalah sama.
- Persoalan mencari jalan keluar dari sebuah Maze
- Persoalan seorang penjual yang akan menjual barangnya ke n buah kota dengan total cost minimum dan setiap kota hanya boleh dikunjungi sekali saja.
- Mencari semua kombinasi dari bilangan 1 sampai n yang jumlahnya adalah n.

12. Berikut adalah kelas persoalan yang masuk dalam kategori NP-Hard, kecuali \*

*Tandai satu oval saja.*

- Sum of Subset Problem
- Halting Decision Problem
- Traveling Salesperson Optimization Problem
- Boolean Satisfiability Problem
- Traveling Salesman Decision Problem

13. Jika pola P = "ritma" akan dicari di teks T = "strategi algoritma" dengan algoritma pencocokan string Boyer-More, berapa jumlah perbandingan karakter yang diperlukan sampai ditemukan pola P tersebut: \*

*Tandai satu oval saja.*

- 7
- 8
- 9
- 10
- 11

14. Misalkan sebuah graf G terdiri dari empat simpul  $V = \{p, q, m, n\}$  dimana setiap simpul terhubung ke simpul lainnya (fully connected). Persoalan Traveling Salesman Problem (TSP) yang berawal dari simpul p dengan Dynamic Programming dimodelkan oleh fungsi rekursif  $f(i, S)$  yang menyatakan bobot lintasan terpendek yang berawal dari simpul i, yang melalui semua simpul di dalam himpunan simpul S dan berakhir pada simpul p. Jika  $C_{ij}$  menyatakan bobot dari simpul i ke simpul j, maka nilai dari  $f(q, \{m,n\})$  pada pendekatan dynamic programming ini dapat dinyatakan dengan:

*Tandai satu oval saja.*

- min {  $C_{qm} + \min \{C_{mn} + f(n, \text{null})\}, C_{qn} + \min \{C_{nm} + f(m, \text{null})\}$  }
- min {  $C_{qm} + \min \{C_{nm} + f(n, \text{null})\}, C_{qn} + \min \{C_{mn} + f(m, \text{null})\}$  }
- min {  $C_{qm} + \min \{C_{mn} + f(m, \text{null})\}, C_{qn} + \min \{C_{nm} + f(n, \text{null})\}$  }
- min {  $C_{mn} + \min \{C_{mq} + f(q, \text{null})\}, C_{mq} + \min \{C_{nq} + f(q, \text{null})\}$  }
- min {  $C_{mn} + \min \{C_{mq} + f(n, \text{null})\}, C_{mq} + \min \{C_{nq} + f(m, \text{null})\}$  }

15. Diberikan regex (regular expression) berikut ini (lihat gambar). Tentukanlah daftar kata yang cocok dengan regex tersebut:

### Regular Expression

**m [ae] n? [dsk]\* [jtiudf]**

*Tandai satu oval saja.*

- (A) mandi, mesti, mati
- (B) mat, mantu, messi
- (C) medi, mansu, maki
- (D) maen, massif, mesi
- (E) mantap, masjid, mensu
- (F) Jawaban A, B, dan C di atas benar
- (G) Tidak ada jawaban yang benar

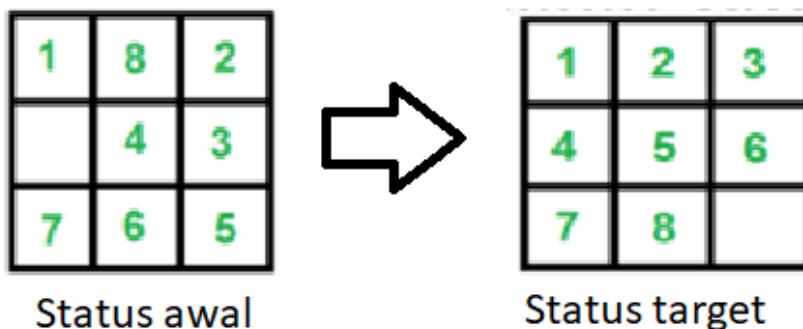
16. Diberikan beberapa pernyataan pada gambar berikut terkait P, NP, dan NP-complete. Tentukan apakah setiap pernyataan benar atau salah.

- (i) Halting Problem adalah contoh sebuah persoalan NP
- (ii) Sebuah persoalan X dikatakan NP-complete jika X termasuk ke dalam kelas NP dan beberapa persoalan di dalam NP lainnya dapat direduksi menjadi instans persoalan X dalam waktu polinomial.
- (iii) Jika A adalah sebuah persoalan di dalam NP-complete dan B adalah persoalan NP tapi tidak perlu NP-complete, maka jika B dapat diselesaikan dalam waktu polinomial maka A juga dapat diselesaikan dalam waktu polinomial
- (iv) Jika persoalan di dalam NP-complete dapat diselesaikan dalam waktu polinomial, maka semua persoalan lain di dalam NP-complete juga dapat diselesaikan dalam waktu polinomial

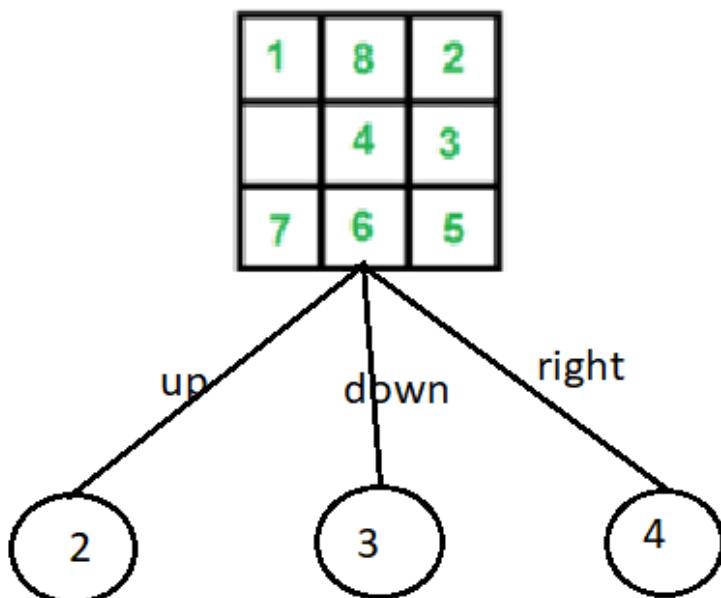
Tandai satu oval saja per baris.

|                  | Benar                            | Salah                            |
|------------------|----------------------------------|----------------------------------|
| Pernyataan (i)   | <input type="radio"/>            | <input checked="" type="radio"/> |
| Pernyataan (ii)  | <input type="radio"/>            | <input checked="" type="radio"/> |
| Pernyataan (iii) | <input type="radio"/>            | <input checked="" type="radio"/> |
| Pernyataan (iv)  | <input checked="" type="radio"/> | <input type="radio"/>            |

17. Diberikan persoalan 8-puzzle berikut ini dan akan diselesaikan dengan algoritma Branch and Bound. Tentukanlah nilai kebenaran setiap pernyataan.



Pohon ruang status:



Tandai satu oval saja per baris.

Benar      Salah

Cost  
simpul  
akar  
adalah 6

Cost  
simpul 2  
adalah 7

Cost

Cost  
simpul 4 \_\_\_\_\_  
adalah 6 \_\_\_\_\_

---

Simpul  
expand \_\_\_\_\_    
berikutnya  
adalah \_\_\_\_\_  
simpul 3 \_\_\_\_\_

---

Nilai  
fungsi \_\_\_\_\_    
Kurang  
untuk  
akar \_\_\_\_\_  
adalah 7 \_\_\_\_\_

18. Terdapat matriks bobot/ matriks jarak seperti pada gambar berikut. Tentukan pernyataan berikut ini benar atau salah terkait solusi persoalan TSP untuk matriks bobot tersebut, dengan pendekatan Branch and Bound. TSP dimulai dari simpul a.

|   | a        | b        | c        | d        | e        |
|---|----------|----------|----------|----------|----------|
| a | $\infty$ | 3        | 1        | 5        | 8        |
| b | 3        | $\infty$ | 6        | 7        | 9        |
| c | 1        | 6        | $\infty$ | 4        | 2        |
| d | 5        | 7        | 4        | $\infty$ | 3        |
| e | 8        | 9        | 2        | 3        | $\infty$ |

Tandai satu oval saja per baris.

Benar      Salah

- a. Jika menggunakan pendekatan matriks bobot tereduksi (reduced cost matrix), maka nilai cost pada akar adalah 13.

b. Cost pada suatu simpul pohon untuk penyelesaian TSP dengan Branch and Bound menunjukkan batas teratas, sehingga ketika didapatkan sebuah solusi pada suatu langkah, simpul lain yang memiliki cost lebih kecil akan 'dimatikan'.

c. Jika pada suatu langkah penyelesaian TSP dengan matriks bobot tereduksi kita memeriksa simpul b (kedalaman 1 pada pohon) yang merupakan anak dari simpul a, maka cost pada simpul b bernilai 13.

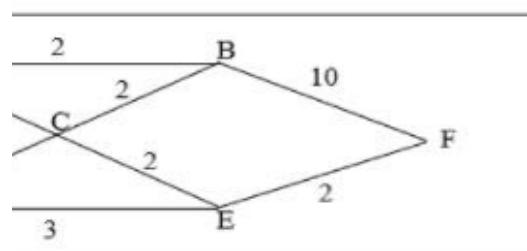
d. Jika penyelesaian TSP menggunakan pendekatan bobot tur lengkap, maka nilai cost pada akar adalah

14.

e. Jika pada suatu langkah penyelesaian TSP dengan bobot tur lengkap kita

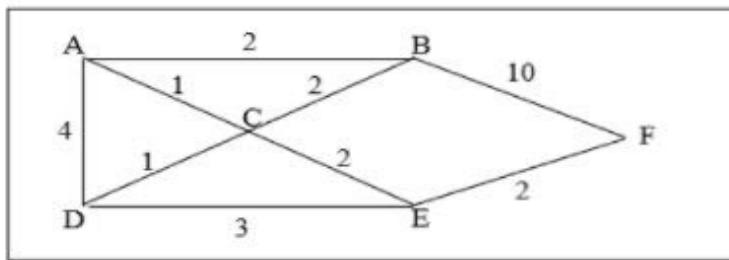
19. <sup>memeriksa</sup> Terdapat sebuah graf tidak berarah berikut ini. Simpul merepresentasikan kota, dan simpul d bilangan yang terdapat pada sisi adalah jarak antara dua kota. Jika nilai heuristik adalah banyaknya sisi minimum dari suatu simpul ke simpul F, maka tentukan nilai heuristik setiap simpul berikut.

merupakan anak dari simpul b (kedalaman 1 pada pohon), dan simpul b merupakan anak dari simpul a (akar), maka



| simpul    | 0                                | 1                                | 2                                | 3                     | 4                     | 5                     | 6                     | 7                     | 8                     |
|-----------|----------------------------------|----------------------------------|----------------------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| a. $h(A)$ | <input type="radio"/>            | <input type="radio"/>            | <input checked="" type="radio"/> | <input type="radio"/> |
| b. $h(B)$ | <input type="radio"/>            | <input checked="" type="radio"/> | <input type="radio"/>            | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| c. $h(C)$ | <input type="radio"/>            | <input type="radio"/>            | <input checked="" type="radio"/> | <input type="radio"/> |
| d. $h(D)$ | <input type="radio"/>            | <input type="radio"/>            | <input checked="" type="radio"/> | <input type="radio"/> |
| e. $h(E)$ | <input type="radio"/>            | <input checked="" type="radio"/> | <input type="radio"/>            | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| f. $h(F)$ | <input checked="" type="radio"/> | <input type="radio"/>            | <input type="radio"/>            | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |

20. Terkait dengan graf pada gambar berikut, pilih pernyataan yang benar di bawah ini.



Centang semua yang sesuai.

- a. Jika nilai heuristik adalah banyaknya sisi minimum dari suatu simpul ke simpul F, maka terdapat simpul yang nilai heuristiknya tidak 'admissible'.
- b. Pendekatan A\* untuk mencari jalur dengan jarak minimum memerlukan iterasi yang paling sedikit.
- c. Pendekatan UCS (Uniform Cost Search) akan menghasilkan jalur A-C-E-F jika dicari jalur dengan jarak (cost) minimum dari kota A ke kota F.
- d. Pendekatan Greedy Best First Search untuk mencari jalur dari A ke F dengan jarak minimum, memerlukan iterasi yang paling sedikit.
- e. Pendekatan Greedy Best First Search dan A\* memerlukan nilai heuristik untuk mencari jalur dengan jarak minimum.
- f. Pendekatan UCS, Greedy Best First Search dan A\* akan menghasilkan jalur yang paling optimal (jarak minimum) dari simpul A ke simpul F.

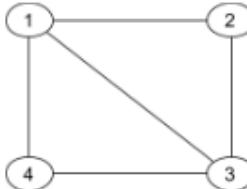
---

Konten ini tidak dibuat atau didukung oleh Google.

Google Formulir

### Bagian I Backtracking dan Route/ Path Planning

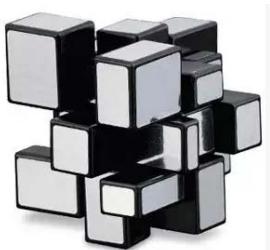
1. Tentukan untuk tiap pernyataan di bawah ini, apakah pernyataan tersebut Benar atau Salah. (Nilai 5)  
(a) Dalam persoalan pewarnaan graf berikut ini, jika diselesaikan dengan Backtracking dan domain warna yang sediakan hanya merah, kuning, dan hijau, maka persoalan tersebut tidak dapat diselesaikan.



- (b) Perhatikan persoalan cryptarithmetic berikut ini (menggantikan setiap huruf dengan angka yang jika dijumlahkan sesuai dengan representasi hurufnya). Jika diselesaikan dengan Backtracking, maka constraint (kendala) yang tidak boleh dilanggar (untuk menentukan apakah fungsi pembatas bernilai true atau false) pada huruf O (pada TWO) dan R (pada FOUR) adalah:  $O + O = R$ .

$$\begin{array}{r} T \quad W \quad O \\ + \quad T \quad W \quad O \\ \hline F \quad O \quad U \quad R \end{array}$$

- (c) Pada persoalan perencanaan rute/ jalur (route/path planning), yang membedakan pendekatan pencarian dengan UCS, Greedy Best First Search, dan A star adalah penentuan simpul berikutnya yang akan diperiksa (diekspan).
  - (d) Penyelesaian persoalan rubic cube dengan initial state pada gambar (a) hingga mendapatkan goal state (gambar b) berikut ini termasuk ke dalam persoalan path planning.



Gambar (a)

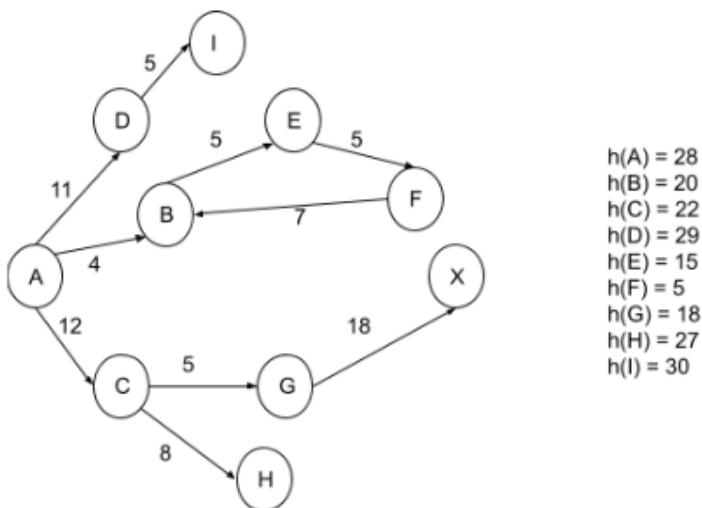


Gambar (b)

- (e) Penyelesaian persoalan route planning dengan pendekatan Greedy Best First Search pasti memberikan hasil optimal karena memanfaatkan heuristik, sedangkan pendekatan UCS belum tentu memberikan hasil optimal karena termasuk blind search.
  2. Persoalan yang diselesaikan dengan Backtracking, perlu didefinisikan kumpulan variabel dan domain dari tiap variabel untuk menyusun solusinya. Untuk contoh instansiasi persoalan Magic Square Puzzle  $n \times n$  dengan  $n=3$ , di mana peletakan angka 1 hingga 9 membuat jumlah dalam satu baris, satu kolom, serta diagonal, sama dengan  $n*(n^2 + 1)/2$ ; sebutkan semua variabel dan domain dari tiap variabelnya. Gambar berikut adalah salah satu contoh solusinya. (Nilai 6)

|   |   |   |
|---|---|---|
| 8 | 1 | 6 |
| 3 | 5 | 7 |
| 4 | 9 | 2 |

3. Terdapat graf berarah sebagai berikut, dan ingin dicari jalur dari simpul A ke simpul X. Graf dilengkapi dengan jarak antar simpul, dan informasi heuristik tiap simpul diberikan.



- (a) Apakah terdapat heuristik yang tidak admissible pada graf tersebut? Jika ada sebutkan heuristik dari simpul mana.(catatan: definisi admissible sesuai dengan salindia kuliah). (4)
- (b) Jika dicari rute dari simpul A ke simpul X dengan A star, dan pencarian dihentikan ketika simpul X menjadi simpul Ekspan, berapa banyak iterasi yang diperlukan? Iterasi 1 saat simpul A menjadi simpul Ekspan. Simpul yang sudah menjadi simpul ekspan tidak dimasukkan lagi ke dalam simpul hidup. Tuliskan jalur yang dihasilkan dan jarak dari jalur tersebut. Jika tidak bisa dihasilkan solusi tuliskan "Tidak ada solusi" pada lembar jawaban. (5)
- (c) Jika dicari rute dari simpul A ke simpul X dengan Greedy Best First Search, dan pencarian dihentikan ketika simpul X menjadi simpul Ekspan, berapa banyak iterasi yang diperlukan? Iterasi 1 saat simpul A menjadi simpul Ekspan. Simpul yang sudah menjadi simpul ekspan tidak dimasukkan lagi ke dalam simpul hidup. Tuliskan jalur yang dihasilkan dan jarak dari jalur tersebut. Jika tidak bisa dihasilkan solusi tuliskan "Tidak ada solusi" pada lembar jawaban. (5)
- (d) Jika dicari rute dari simpul A ke simpul X dengan Uniform Cost Search (UCS), dan pencarian dihentikan ketika simpul X menjadi simpul Ekspan, berapa banyak iterasi yang diperlukan? Iterasi 1 saat simpul A menjadi simpul Ekspan. Simpul yang sudah menjadi simpul ekspan tidak dimasukkan lagi ke dalam simpul hidup. Tuliskan jalur yang dihasilkan dan jarak dari jalur tersebut. Jika tidak bisa dihasilkan solusi tuliskan "Tidak ada solusi" pada lembar jawaban. (5)

## Bagian II Branch & Bound dan String Matching

1. (Nilai 10) Diberikan persoalan n-puzzle berikut ini:

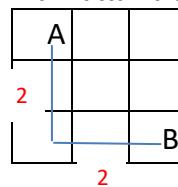
State awal

|   |   |   |
|---|---|---|
| 1 | 2 | 3 |
| 7 | 8 | 4 |
| 6 |   | 5 |

Goal

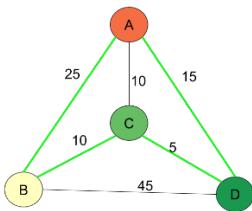
|   |   |   |
|---|---|---|
| 1 | 2 | 3 |
| 8 |   | 4 |
| 7 | 6 | 5 |

Manhattan distance A ke B =  $2+2=4$



- a. (Nilai 2) Jelaskanlah apakah goal bisa dicapai dari state awal yang diberikan.
- b. (Nilai 6+2) Didefinisikan fungsi heuristik akan mengembalikan nilai heuristik berupa *jumlah jarak setiap tile (kecuali tile kosong) ke goal position*, dan cost dari akar ke suatu state adalah *jumlah step dari akar ke state tersebut*. Jarak yang digunakan adalah Manhattan distance, contoh distance untuk tile 8 adalah 1. Berikanlah pohon ruang status pencarian solusi n-puzzle ini dengan selalu menggunakan urutan aksi LRUD dan pohon hanya berisi state yang unik. Lengkapilah dengan nomor simpul dan perhitungan cost, dimulai dari akar yang diberi nomor nol. Berikanlah solusi berupa urutan aksinya di akhir jawaban. Jika dilakukan pruning, tulislah secara eksplisit sebagai keterangan.

2. (Nilai 6) Diberikan persoalan TSP dengan simpul awal A sbb. Pohon ruang status pencarian solusi TSP dibentuk dengan menambahkan simpul dimulai dari simpul 1 untuk akar.



- a. (Nilai 3) Jelaskan cara perhitungan cost dengan reduced cost matriks untuk simpul 3 yang merepresentasikan lintasan A-C, mulailah dengan membentuk matriks berbobot dengan urutan simpul ABCD
- b. (Nilai 3) Jelaskan cara perhitungan cost dengan bobot tur lengkap untuk simpul 3 yang merepresentasikan lintasan A-C.
3. (Nilai 4) Diberikan persoalan Integer knapsack berikut ni.  $K=6$ .  $(w_1, w_2, w_3, w_4) = (3, 2, 2, 1)$ ;  $(p_1, p_2, p_3, p_4) = (35, 80, 60, 95)$ . Jika diselesaikan dengan Branch & Bound, bentuklah **pohon ruang status pencarian solusi** sampai dengan level 1 saja dilengkapi dengan perhitungan costnya.
4. (Nilai 15) Diberikan sebuah teks: **work carefully in lynwood**
- a. (Nilai 3) Jelaskanlah apakah metode KMP atau Boyer-Moore yang lebih baik digunakan dalam proses pencocokan *string* dengan pola **ly** terhadap teks tersebut.
- b. (Nilai 7) Aplikasikanlah metode yang dianggap lebih baik pada soal (a) dengan menggambarkan proses pencocokan *string* sampai pola **ly** ditemukan atau sudah di akhir teks. Berapa jumlah perbandingan karakter yang terjadi?
- c. (Nilai 5) Apakah hasil dari fungsi **re.findall(r" [\w\s]+ly\b", teks)**. Jelaskanlah terlebih dahulu pola yang direpresentasikan oleh regex tersebut.

### Bagian III Dynamic Programming dan Teori P, NP, NPC

#### A. Program Dinamis

Misalkan di sisi kiri jalan sebuah kompleks perumahan terdapat  $n$  buah rumah. Setiap rumah ke- $i$  memiliki nilai kekayaan di dalamnya sebesar  $p_i$ . Sorang pencuri akan mencuri harta dari rumah-rumah tersebut. Namun dia tidak dapat mencuri pada dua rumah yang bertetangga (baik tetangga sebelah kiri maupun tetangga sebelah kanan). Berapa nilai maksimum harta yang dapat dia curi?

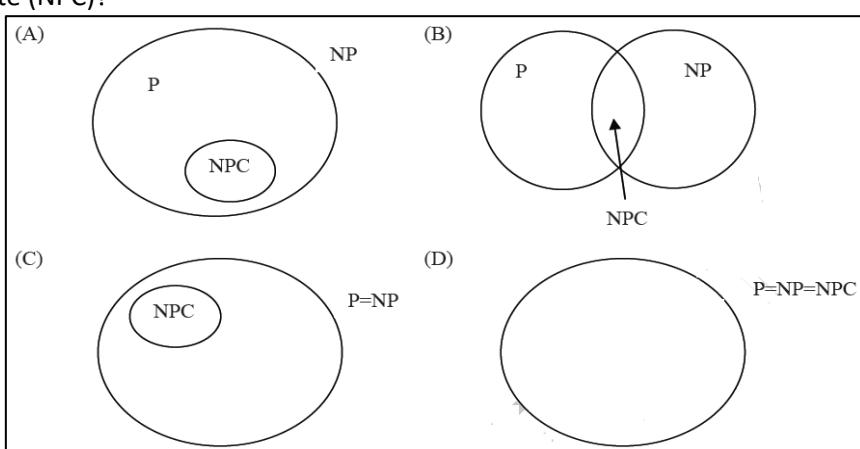
- (a) Misalkan  $n = 7$  dan  $p = [9, 3, 5, 8, 2, 4, 7]$ . Jika diselesaikan secara *brute force*, berapa maksimum nilai kekayaan yang dapat dia peroleh? (5)
- (b) Jika diselesaikan dengan program dinamis, tentukan relasi (persamaan rekursif) yang memperlihatkan keuntungan yang diperoleh oleh pencuri (basis dan rekurens) nya (10)
- (c) Selesaikan perhitungan untuk jawaban b dengan contoh instans persoalan seperti pada pertanyaan (a) di atas. (5)

#### B. Teori P, NP, dan NP-Complete

Pilihlah satu jawaban yang benar (tulis jawaban pada lembar jawaban). Setiap soal bernilai 2. (total  $2 \times 7 = 14$ )

1. Persoalan yang dapat dipecahkan dalam polinom dikenal sebagai
  - tractable
  - intractable
  - decision
  - unsolvable
  - complete
2. Algoritma non-deterministik dikatakan *non-deterministic polynomial algorithm* jika kompleksitas waktu tahap menerkanya adalah polinom.
  - Benar
  - Salah

3. Berapa tahap yang dibutukan untuk membuktikan sebuah persoalan keputusan termasuk NP-complete?
- 1
  - 2
  - 3
  - 4
  - 5
4. Persoalan keputusan mana dibawah ini yang tidak termasuk ke dalam NP-complete?
- Sirkuit Hamilton
  - Bin packing problem
  - Partition problem
  - Minesweeper problem
  - Halting problem
5. Dengan mengasumsikan  $P \neq NP$ , pernyataan manakah dari berikut ini yang benar?
- $NP\text{-complete} = NP$
  - $NP\text{-complete} \cap P = \emptyset$
  - $NP\text{-hard} = NP$
  - $P = NP\text{-complete}$
  - Tidak ada yang benar
6. Manakah dari ketiga pernyataan ini yang BENAR?
- Persoalan menentukan apakah terdapat sirkuit di dalam sebuah graf tak-berarah adalah persoalan P
  - Persoalan menentukan apakah terdapat sirkuit di dalam sebuah graf tak-berarah adalah persoalan NP
  - Jika persoalan A adalah NP-complete, maka terdapat *non-deterministic polynomial time algorithm* untuk menyelesaikan A.
- 1, 2 dan 3
  - 1 dan 3
  - 2 dan 3
  - 1 and 2
  - 1 saja
7. Misalkan algoritma dengan kebutuhan waktu polinom ditemukan untuk menyelesaikan persoalan TSP untuk graf yang besar. Dengan skenario ini, manakah dari diagram Venn berikut yang merepresentasikan kelas P, NP, dan NP-Complete (NPC)?



### C. Bagian Perkiraan Nilai

Tuliskan perkiraan nilai anda untuk mata kuliah ini (A/AB/B/BC/C/D/E)

(Nilai: 2)

Ujian Akhir Semester IF2211 Strategi Algoritma

Jumat 19 Mei 2023

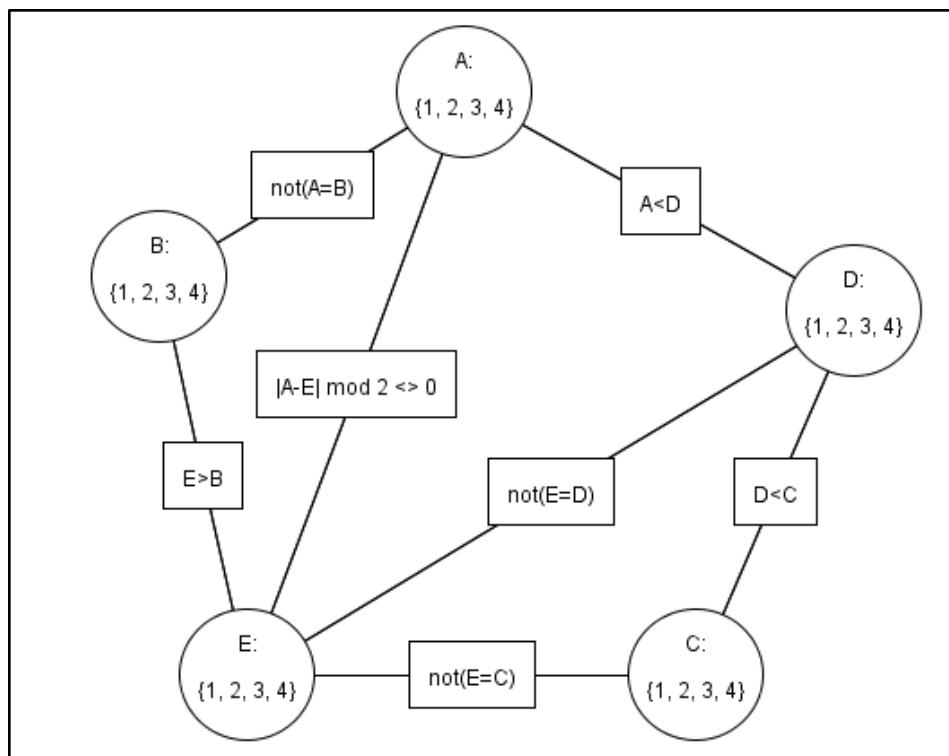
Waktu: 120 menit

Dosen: Rinaldi Munir, Nur Ulfa Maulidevi, Rila Mandala

Berdoalah terlebih dahulu agar Anda sukses dalam ujian ini!

**Bagian Backtracking, UCS, Greedy Best First Search, A\***

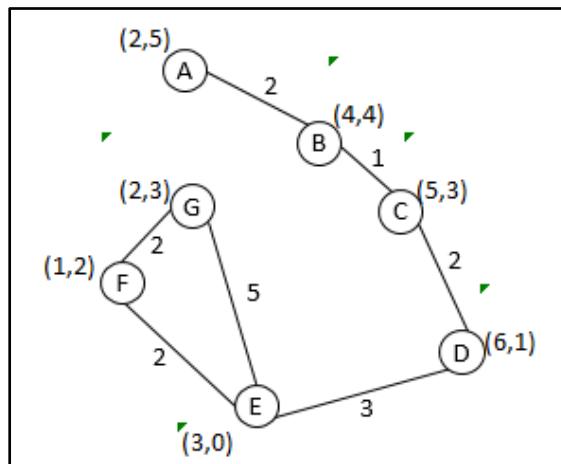
1. Terdapat persoalan penjadwalan yang diilustrasikan pada Gambar 1. Lima kegiatan (A,B,C,D,E) akan dijadwalkan waktu mulainya pada satuan waktu 1, 2, 3, atau 4. Proses penjadwalan tidak boleh melanggar batasan yang dituliskan pada Gambar 1. Solusi penjadwalan dicari dengan pendekatan Backtracking. **(Nilai 10)**



Gambar 1. Penjadwalan Lima Kegiatan

- a. Tuliskan representasi solusi dari persoalan penjadwalan tersebut (variabel apa saja, dan domain atau nilai yang mungkin diisikan pada tiap variabel).  
b. Tuliskan batasan yang tidak boleh dilanggar saat pencarian solusi penjadwalan pada Gambar 1.  
c. Tentukan apakah persoalan penjadwalan pada Gambar 1 dapat diselesaikan atau tidak. Jika tidak dapat diselesaikan, jelaskan alasannya. Jika dapat diselesaikan, tuliskan solusi persoalan penjadwalan tersebut.
2. Terdapat sebuah graf tidak berarah seperti pada Gambar 2. Dalam Gambar 2 diberikan posisi dari tiap simpul, dan cost (biaya) kebutuhan bahan bakar antar simpul. Carilah rute/ jalur **dari simpul C menuju ke simpul G**, dengan beberapa teknik berikut. Pencarian dihentikan (iterasi dihentikan) jika sudah sampai simpul G (simpul G menjadi simpul ekspan). Simpul yang sudah pernah diperiksa (di-ekspan), tidak akan diperiksa lagi (tidak masuk agenda). Jika diperlukan, heuristik yang digunakan

adalah **manhattan distance** antara dua titik. Jika dalam proses pencarian dua buah simpul memiliki nilai evaluasi yang sama, maka prioritas simpul yang diperiksa sesuai urutan abjad. (**Nilai 20**)



Gambar 2. Graf Tidak Berarah

- Tuliskan nilai *cost* (biaya) yang diperlukan dari simpul C ke setiap simpul yang lain. Jika dimungkinkan untuk melalui lebih dari 1 jalur (path) untuk menuju suatu simpul dari simpul C, tuliskan *cost* untuk setiap jalur yang mungkin. Cara penulisan jawaban:  $g(A) = \dots$  {titik-titik diisi dengan *cost* dari simpul C menuju simpul A}.
- Tuliskan heuristik dari setiap simpul untuk menuju ke simpul G; dan tentukan apakah heuristik tersebut *admissible* atau tidak. Cara penulisan jawaban:  $h(A) = \dots$  / Tidak *admissible* karena [alasan] {titik-titik diisi dengan nilai heuristik, dan misal heuristik simpul A tidak *admissible* maka dituliskan "Tidak *admissible*" beserta alasannya}.
- Tuliskan formula masing-masing untuk menghitung nilai evaluasi setiap simpul  $f(n)$ , dengan memanfaatkan  $g(n)$  dan  $h(n)$  pada soal (a) dan soal (b) untuk pendekatan UCS, Greedy Best First Search, dan A\*.
- Tuliskan jalur yang dihasilkan serta total *cost* yang diperlukan untuk setiap pendekatan UCS, Greedy Best First Search, dan A\*. Analisis dengan singkat, pendekatan yang terbaik dilihat dari aspek hasil jalur dengan *cost* termurah dan banyaknya iterasi pencarian yang diperlukan.

### Bagian Programma dinamis

- Sebuah perusahaan modifikasi/*repairing* mobil untuk melakukan modifikasi/*repairing* mobil harus melalui 4 tahapan sebagai berikut : (1)*Towing*, (2)*Inspection and Diagnostic*, (3)*Disassembling and Repair*, dan (4)*Reassembling and Testing*. Untuk setiap tahapan tersebut, perusahaan tersebut mempunyai 4 *station* yang berbeda jarak dan unit *cost* dari modifikasi/*repairing* mobil tersebut. Biaya per unit mobil yang diperbaiki dari satu *station* ke *station* lain seperti di bawah ini :

| Dari Towing | Ke Inspection and Diagnostic Station |    |    |    |
|-------------|--------------------------------------|----|----|----|
|             | 1                                    | 2  | 3  | 4  |
|             | 35                                   | 40 | 30 | 45 |

| Dari Inspection and Diagnostic Station | Ke Disassembling and Repair Station |     |     |     |
|--|-------------------------------------|-----|-----|-----|
|  | 1                                   | 2   | 3   | 4   |
| 1                                      | 105                                 | 100 | 85  | 90  |
| 2                                      | 90                                  | 85  | 100 | 95  |
| 3                                      | 100                                 | 90  | 95  | 105 |
| 4                                      | 110                                 | 105 | 120 | 110 |

| Dari Disassembling and Repair Station | Ke Reassembling and Testing Station |    |    |    |
|---------------------------------------|-------------------------------------|----|----|----|
|                                       | 1                                   | 2  | 3  | 4  |
| 1                                     | 70                                  | 75 | 85 | 80 |
| 2                                     | 85                                  | 90 | 80 | 95 |
| 3                                     | 90                                  | 70 | 85 | 80 |
| 4                                     | 80                                  | 85 | 90 | 75 |

Tentukanlah penjadwalan yang paling optimal beserta biaya pada masing-masing station serta total biaya yang harus dikeluarkan dengan menggunakan *Dynamic Programming*. **(Nilai 15)**

### **Bagian Teori P, NP, NP Complete**

4. Tuliskan apakah soal di bawah ini benar atau salah :
- Persoalan Menara Hanoi adalah termasuk contoh persoalan *NP-complete problem* **(Nilai 4)**
  - Persoalan Menara Hanoi adalah tidak termasuk contoh persoalan *NP-Hard* **(Nilai 4)**
  - Persoalan *Fast Fourier Transform* adalah termasuk ke dalam persoalan NP-complete. **(Nilai 4)**
  - Persoalan *Halting-problem* adalah satu-satunya persoalan yang *un-solved*. **(Nilai 4)**
  - Persoalan *post correspondent problem* yaitu bagaimana menyusun *list* dari pasangan dua buah string dapat disusun dengan cara susunan tertentu adalah termasuk persoalan *intractable*. **(Nilai 4)**

### **Bagian Algoritma Branch and Bound dan String Matching dan Regular Expression**

5. Lima buah proyek pembuatan perangkat lunak akan ditugaskan kepada lima buah tim. Setiap tim akan mengerjakan satu proyek saja dan setiap proyek hanya dikerjakan oleh satu buah tim. Tidak ada tim yang mengerjakan proyek sekaligus. Setiap tim punya kemampuan *programming* dan *coding* yang berbeda-beda, sehingga lama waktu pengerjaan proyek oleh setiap tim tidak sama. Matriks berikut memperlihatkan estimasi waktu (dalam satuan hari) yang dibutuhkan oleh setiap tim untuk mengerjakan proyek. Gunakan algoritma *branch and bound* untuk menentukan bagaimana cara menugaskan tim dengan proyek sehingga total waktu pengerjaan seluruh proyek adalah seminimal mungkin. Dalam menjawab soal ini, perlihatkan pembangunan pohon ruang statusnya, *bound (cost)* setiap simpul, simpul yang dibunuh, sampai menemukan *goal node*. **(Nilai 18)**

|       | Proyek A | Proyek B | Proyek C | Proyek D | Proyek E |
|-------|----------|----------|----------|----------|----------|
| Tim 1 | 9 hari   | 4 hari   | 6 hari   | 9 hari   | 8 hari   |
| Tim 2 | 8 hari   | 6 hari   | 5 hari   | 2 hari   | 7 hari   |
| Tim 3 | 5 hari   | 8 hari   | 3 hari   | 4 hari   | 4 hari   |
| Tim 4 | 7 hari   | 6 hari   | 10 hari  | 3 hari   | 7 hari   |
| Tim 5 | 5 hari   | 9 hari   | 8 hari   | 7 hari   | 2 hari   |

6. (a) Berikan contoh sebuah *pattern* sepanjang 5 karakter dan teks sepanjang  $> 10$  karakter sedemikian sehingga algoritma pencocokan string dengan KMP sama jumlah perbandingan karakternya dengan algoritma *brute force* pada kasus terburuk. Perlihatkan proses pencocokannya dan jumlah perbandingan karakter pada masing-masing algoritma.
- (b) Diberikan teks sebagai berikut: WELCOME TO MY COAL LISION. Carilah pattern COAL dengan algoritma Boyer-Moore. Dalam menjawab soal ini, perlihatkan proses pencocokan stringnya, hitung *last occurrence*, dan hitung jumlah perbandingan karakter yang terjadi
- (c) Diberikan kode *regex* sebagai berikut: `[A-Za-z]{2}\d{4}` dan sebuah teks sbb:  
 IFangkatan2021NIM13521xyzUjianIF2211padatanggal19Mei2023  
 Tentukan semua string yang cocok dengan *regex* tersebut **(Nilai 6 + 6 + 5)**
7. Tentukan prediksi nilai anda untuk kuliah ini (A/AB/B/BC/C/D/E) **(Nilai = 2)**

Ujian Akhir Semester **IF2211 Strategi Algoritma**

Kamis, 6 Juni 2024

Waktu: 180 menit

Dosen: Rinaldi Munir, Nur Ulfa Maulidevi, Rila Mandala/Monterico Adrian

*Berdoalah terlebih dahulu agar Anda sukses dalam ujian ini!*

**Algoritma Runut-balik (*backtracking*)**

1. Terdapat persoalan *cryptarithmetic* berikut ini. (**Nilai 15**)

$$\begin{array}{r} T \quad W \quad O \\ + \quad T \quad W \quad O \\ \hline F \quad O \quad U \quad R \end{array}$$

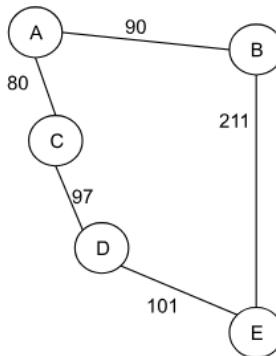
Setiap huruf harus diganti dengan sebuah angka (huruf yang sama merepresentasikan angka yang sama, huruf yang berbeda harus merepresentasikan angka yang berbeda), sedemikian sehingga ketika sudah dalam bentuk rangkaian angka, ekspresi matematika menunjukkan penjumlahan yang benar. Huruf pertama pada setiap kata tidak boleh diganti dengan angka nol.

Dengan memanfaatkan algoritma *Backtracking* untuk persoalan *cryptarithmetic* tersebut, tentukan:

- Representasi dari solusi persoalan
- Fungsi Pembangkit
- Batasan-batasan (*constraints*) dari persoalan tersebut agar solusi yang diberikan benar.

Catatan: Jawaban untuk butir (a) hingga butir (c) tidak perlu sampai menyelesaikan (menghasilkan solusi) persoalan *cryptarithmetic* tersebut."

2. Terdapat graf tidak berarah dengan jarak antar simpul sebagai berikut.



Dari graf tersebut ingin dicari jalur terpendek dari simpul A ke simpul E. Jika diperlukan, heuristik suatu simpul n atau  $h(n)$  dari persoalan tersebut adalah banyaknya sisi/ busur minimal yang menghubungkan simpul n dengan simpul tujuan. Pencarian solusi menggunakan Uniform Cost Search (UCS), Greedy Best First Search, atau A\*; dengan iterasi pertama untuk semua teknik pencarian adalah sebagai berikut.

| Iterasi | Simpul Ekspan | Simpul Hidup |
|---------|---------------|--------------|
| 1       | A             | B, C         |

Proses pencarian dihentikan ketika simpul tujuan sudah menjadi simpul ekspan. (**Nilai 15**)

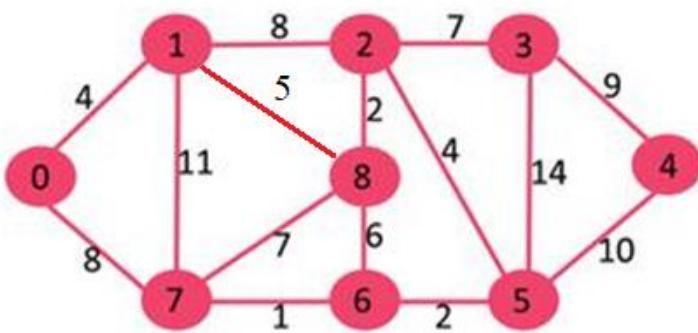
- Tentukan nilai heuristik untuk semua simpul pada graf, dan sebutkan untuk setiap simpul apakah nilai heuristiknya *admissible* atau tidak.
- Pada iterasi ke-4, sebutkan untuk setiap teknik pencarian (3 teknik di soal), simpul apa yang menjadi simpul ekspan.
- Pada iterasi ke-4, sebutkan untuk setiap teknik pencarian (3 teknik di soal), simpul apa yang menjadi simpul hidup dan nilai  $f(n)$  untuk setiap simpul hidup pada setiap teknik pencarian.
- Untuk setiap teknik pencarian, sebutkan pada iterasi ke berapa proses pencarian dihentikan.
- Untuk setiap teknik pencarian, tuliskan jalur hasil pencarian dan total jaraknya.

#### Algoritma Branch and Bound

- Diketahui pada sebuah pesawat ulang alik dapat membawa jet booster seberat 150 satuan. Terdapat 5 tipe jet: Jet A yang mempunyai berat 15 dan kemampuan *boost* 14, Jet C yang mempunyai berat 67 dan kemampuan *boost* 32, Jet E yang mempunyai berat 8 dan kemampuan *boost* 4, Jet G yang mempunyai berat 75 dan kemampuan *boost* 52, dan Jet K yang mempunyai berat 30 dan kemampuan *boost* 7. Tentukan jet mana saja yang dibawa yang akan memberikan *boosting* maksimum dengan algoritma *Branch and Bound*. **(Nilai 10)**

#### Bagian Programma dinamis

- Seekor semut berjalan dari titik 0 ke titik 4. Tentukan rute terpendek yang dilalui oleh semut itu dengan algoritma programma dinamis (*dynamic programming*) dan panjang rute. Dalam menjawab soal ini harus menggunakan tabel perhitungan pada setiap tahap dan rekonstruksi solusi pada akhir tahap. **(Nilai 15)**



#### Bagian Teori P, NP, NP Complete

- Tuliskan apakah soal di bawah ini **benar** atau **salah** (setiap soal bernilai 1,5):
  - Persoalan "diberikan sebuah matriks M berukuran  $n \times n$ , tentukan apakah M memiliki nilai eigen" termasuk ke dalam kelas *NP*.
  - Persoalan *minimum spanning tree* termasuk ke dalam kelas *P*.
  - Jika X adalah dua persoalan di dalam kelas *NP-complete*, Z adalah persoalan di dalam kelas NP tetapi bukan *NP-complete*, maka jika suatu hari nanti Z dapat diselesaikan dalam waktu polinom, maka X juga dapat diselesaikan dalam waktu polinom.
  - Dengan mengasumsikan  $P \neq NP$ , maka *NP-complete* = *P*
  - Persoalan *SAT* (*satisfiable problem*) termasuk ke dalam kelas *NP-complete*
  - Diberikan persoalan *SAT* dengan dua peubah,  $a$  dan  $b$ . Fungsi  $C = a \wedge \bar{b}$  adalah *satisfiable*.
  - Diberikan persoalan *SAT* dengan dua peubah,  $a$  dan  $b$ . Fungsi  $C = a \wedge \bar{a} \wedge b$  tidak *satisfiable*.
  - Algoritma non-deterministik memiliki tahap verifikasi dalam waktu non-polinomial
  - Kelas *P* adalah himpunan bagian dari *NP* karena *P* tidak memiliki tahap verifikasi

- j) Persoalan “P versus NP” adalah persoalan keputusan yang waktu penyelesaiannya non-polinomial namun solusinya dapat diverifikasi dalam waktu polinomial.

### Pencocokan string dan regex

6. Untuk algoritma KMP, diketahui substring DNA sbb : GCAGAGAG. (Nilai 10)
- Buatlah table border function nya.
  - Jika diketahui urutan string DNA sbb : GCATCGCAGAGAGTATACTACGTACG, ilustrasikan Langkah-langkah perbandingan karakter per karakter di algoritma KMP untuk mencari substring di atas pada string tersebut untuk mencari jumlah operasi perbandingan character yang dilakukan.
  - Berapakah jumlah operasi perbandingan character nya total ?
7. Untuk algoritma Boyer Moore, diketahui substring DNA sbb : GCAGAGAG. (Nilai 10)
- Buatlah table Last-occurrence function (L()) nya.
  - Jika diketahui urutan string DNA sbb : GCATCGCAGAGAGTATACTACGTACG, ilustrasikan Langkah-langkah perbandingan karakter per karakter di algoritma Boyer-Moore untuk mencari substring di atas pada string tersebut untuk mencari jumlah operasi perbandingan character yang dilakukan.
  - Berapakah jumlah operasi perbandingan character nya total ?
8. Ubahlah menjadi sebuah REGEX dengan notasi sesuai dengan yang di slide untuk finite automata sbb :
- Finite automata
- 
- ```

graph LR
    Start((Start)) --> R((R))
    R -- R --> R
    R -- S --> S((S))
    S -- T --> R
    S -- U --> U(((U)))
    U -- U --> U
  
```
- Finite automata dengan epsilon sbb :
- 
- ```

graph LR
    Start((Start)) -- ε --> S1(( ))
    Start -- ε --> S2(( ))
    Start -- ε --> S3(( ))
    S1 -- ε --> S4(( ))
    S1 -- ε --> S5(( ))
    S2 -- ε --> S6(( ))
    S2 -- ε --> S7(( ))
    S3 -- ε --> S8(( ))
    S4 -- 0 --> S9(( ))
    S5 -- 0 --> S9
    S6 -- 1 --> S10(( ))
    S7 -- 1 --> S10
    S9 -- ε --> S11(( ))
    S9 -- ε --> S12(( ))
    S10 -- ε --> S13(( ))
    S10 -- ε --> S14(( ))
    S11 -- ε --> S15(((Final)))
    S12 -- ε --> S15
    S13 -- ε --> S15
    S14 -- ε --> S15
  
```
- (Nilai 10)
9. Diberikan regex (regular expression)  $m[a\epsilon]n?[dsk]^*[jtiudf]$  . Tentukanlah kata yang cocok dengan regex tersebut pada himpunan kata berikut:
- mandi, mesti, mati, mat, mantu, messi, medi, mansu,  
maki, maen, massif, mesi, mantap, masjid, mensu
- (Nilai 5)
10. Tentukan prediksi nilai anda untuk kuliah ini (A/AB/B/BC/C/D/E) (Nilai = 2)

**Total nilai = 107**