Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung

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') = (selisih x' dengan posisi x titik Goal) + (selisih y' dengan posisi y titik Goal)

4	S			
3				
2				
1				G
	1	2	3	4

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.

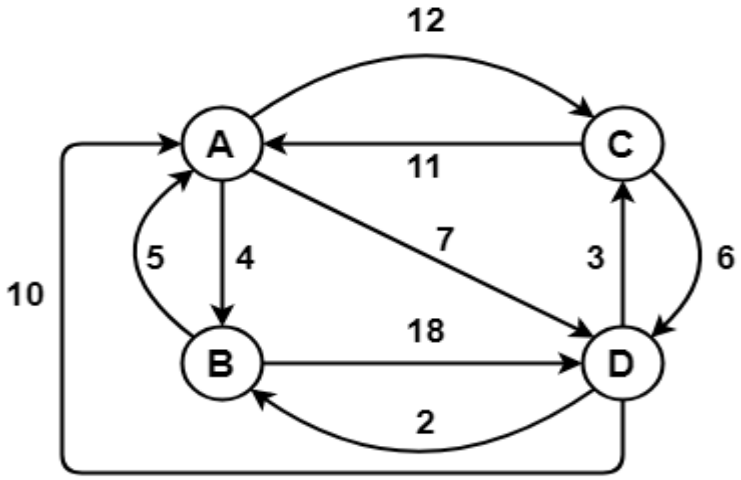
b. (**Nilai 7**) Tentukan nilai heuristik dari setiap titik yang bisa dilewati pada gambar 1, dan sebutkan titik-titik dengan nilai heuristik yang tidak admissible jika ada.

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) = … {Isikan formula untuk mencari f(n) dengan UCS}		Formula: f(n) = … {Isikan formula untuk mencari f(n) dengan Greedy Best First Search}		Formula: f(n) = … {Isikan formula untuk mencari f(n) dengan A Star}	
	Titik - Ekspan	Titik Hidup (tuliskan semua titik hidup dan nilai f(n) untuk tiap titik hidup)	Titik - Ekspan	Titik Hidup (tuliskan semua titik hidup dan nilai f(n) untuk tiap titik hidup)	Titik - Ekspan	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) - … - … Jarak = …		Jalur = (1,4) - … - … Jarak = …		Jalur = (1,4) - … - … Jarak = …	

Bagian B (Branch&Bound dan Dynamic Programming)

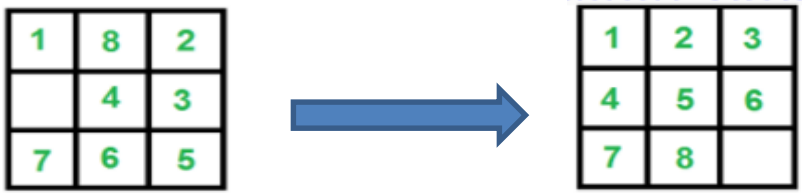
2. Persoalan TSP berikut meminimumkan jarak sirkuit hamilton, dengan simpul awal = A.



(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 = 00000000000000001. 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* adalan 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)