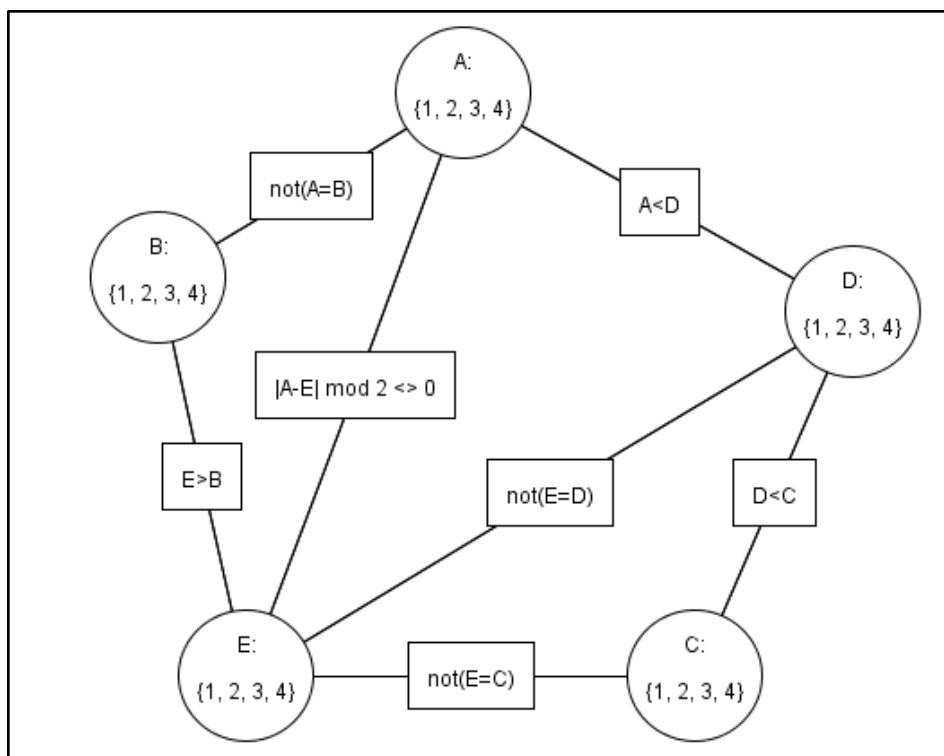


Berdoalah terlebih dahulu agar Anda sukses dalam ujian ini!

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

- Tuliskan representasi solusi dari persoalan penjadwalan tersebut (variabel apa saja, dan domain atau nilai yang mungkin diisikan pada tiap variabel).
- Tuliskan batasan yang tidak boleh dilanggar saat pencarian solusi penjadwalan pada Gambar 1.
- 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.

Jawaban

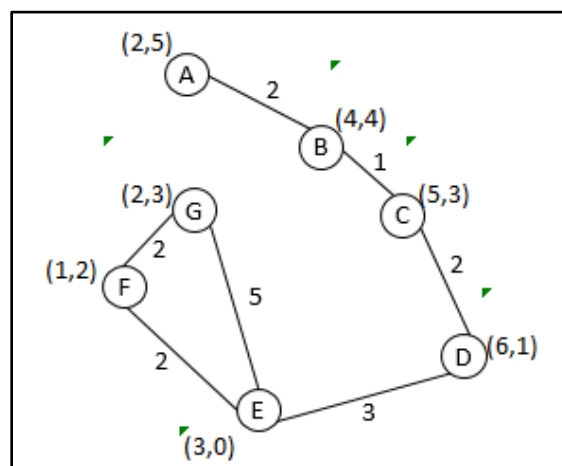
- a. Representasi Solusi: tuple $X(x_1, x_2, x_3, x_4, x_5)$ atau tuple $X(A, B, C, D, E)$ dengan domain tiap variabel adalah $\{1, 2, 3, 4\}$.
- b. Batasan:
 - $A < > B$
 - $A < D$

- (iii) $|A-E| \text{ mod } 2 \neq 0$
- (iv) $E < B$
- (v) $E \neq C$
- (vi) $D < C$
- (vii) $E \neq D$

c. Dapat diselesaikan.

Salah satu solusinya: $X(1, 2, 3, 2, 4)$ {urutan sesuai dengan kegiatan A,B,C,D,E}

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

- a. 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}.
- b. 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}.
- c. 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*.
- d. 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.

Jawaban

a. Tabel g(n)

| Simpul | g(n) | |
|--------|------|----------------|
| A | 3 | |
| B | 1 | |
| C | 0 | |
| D | 2 | |
| E | 5 | |
| F | 7 | jika C-D-E-F |
| | 12 | jika C-D-E-G-F |
| G | 10 | jika C-D-E-G |
| | 9 | jika C-D-E-F-G |

b. Tabel h(n) dan admissibility

| Simpul | h(n) | Admissible? |
|--------|------|-------------------------|
| A | 2 | Admissible, $2 \leq 13$ |
| B | 3 | Admissible, $3 \leq 11$ |
| C | 3 | Admissible, $3 \leq 10$ |
| D | 6 | Admissible, $6 \leq 8$ |
| E | 4 | Admissible, $4 \leq 5$ |
| F | 2 | Admissible, $2 \leq 2$ |
| G | 0 | Admissible, $0 \leq 0$ |

c. Formula

UCS: $f(n) = g(n)$

Greedy Best First Search: $f(n) = h(n)$

A*: $f(n) = g(n) + h(n)$

d. Solusi

UCS: Jalur = C – D – E – F – G, cost: 9

Greedy Best First Search: Jalur = C – D – E – G, cost = 10

A*: Jalur = C – D – E – F – G, cost = 9

Analisis minimal berisi: cost termurah dihasilkan oleh UCS dan A*, namun iterasi minimal dihasilkan oleh Greedy.

Rincian pencarian:

| Iterasi | UCS | | Greedy Best First Search | | A* | |
|---------|-----------------|---|--------------------------|---|-----------------|---|
| | Simpul Ekspansi | Simpul Hidup dan nilai f(n) untuk tiap simpul | Simpul Ekspansi | Simpul Hidup dan nilai f(n) untuk tiap simpul | Simpul Ekspansi | Simpul Hidup dan nilai f(n) untuk tiap simpul |
| 1 | C | B(1) D(2) | C | B(3) D(6) | C | B(4) D(8) |
| 2 | B | D(2) A(3) | B | A(2) D(6) | B | A(4) D(8) |

| | | | | | | |
|------------------------|-------------------|---|---------------|--------------|-------------------|---|
| 3 | D | A(3) E(5) | A | D(6) | A | D(8) |
| 4 | A | E(5) | D | E(4) | D | E(9) |
| 5 | E | F(7) G(10) | E | G(0) F(2) | E | F(9) G(10) |
| 6 | F | G _F (9) G _E (10) | G | Sudah goal | F | G _F (9) G _E (10) |
| 7 | G _F | Sudah goal | | | G _F | Sudah goal |
| Jalur yang dihasilkan: | C – D – E – F – G | | C – D – E – G | | C – D – E – F – G | |
| Total Cost jalur: | 9 | | 10 | | 9 | |

Bagian Programma dinamis

3. 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)**

Jawaban:

Dengan pendekatan pemrograman dinamis metode *backward*, maka harus di mulai pada stage-1 dan hasilnya seperti pada Tabel di bawah ini :

| $S_1 \backslash X_1$ | $f_1(S_1, X_1) = c_{s_1, x_1}$ | | | | $f_1^*(S_1)$ | x_1^* |
|----------------------|--------------------------------|----|----|----|--------------|---------|
| | 1 | 2 | 3 | 4 | | |
| 1 | 70 | 75 | 85 | 80 | 70 | 1 |
| 2 | 85 | 90 | 80 | 95 | 80 | 3 |
| 3 | 90 | 70 | 85 | 80 | 70 | 2 |
| 4 | 80 | 85 | 90 | 75 | 75 | 4 |

Kemudian dilanjutkan pada stage-2 dan hasilnya seperti pada Tabel di bawah ini :

| $S_2 \backslash X_2$ | $f_2(S_2, X_2) = c_{s_2, x_2} + f_1^*(S_1)$ | | | | $f_2^*(S_2)$ | x_2^* |
|----------------------|---|-------------------|-------------------|-------------------|--------------|---------|
| | 1 | 2 | 3 | 4 | | |
| 1 | 105 + 70 = 175 | 100 + 80 = 180 | 85 + 70 = 155 | 90 + 75 = 165 | 155 | 3 |
| 2 | 90 + 70 = 160 | 85 + 80 = 165 | 100 + 70 = 170 | 95 + 75 = 170 | 160 | 1 |
| 3 | 100 + 70 = 170 | 90 + 80 = 170 | 95 + 70 = 165 | 105 + 75 = 180 | 165 | 3 |
| 4 | 110 + 70 = 180 | 105 + 80 = 185 | 120 + 70 = 190 | 110 + 75 = 185 | 180 | 1 |

Kemudian dilanjutkan pada stage-3 dan hasilnya seperti pada Tabel di bawah ini :

| $S_3 \backslash X_3$ | $f_3(S_3, X_3) = c_{s_3, x_3} + f_2^*(S_2)$ | | | | $f_3^*(S_3)$ | x_3 |
|----------------------|---|-------------------|-------------------|-------------------|--------------|-------|
| | 1 | 2 | 3 | 4 | | |
| Dari Towing | 35 + 155 = 190 | 40 + 160 = 200 | 30 + 165 = 195 | 45 + 180 = 225 | 190 | 1 |

stage-3 → stage-2 → stage-1.

Dari Towing ke R&T Station melalui mesin-1, dengan biaya = 35. Dari R&T Station di mesin-1 ke D&R Station melalui mesin-3 dengan biaya = 85. Dari D&R Station di mesin-3 ke I&D Station melalui mesin-2 dengan biaya = 70. Total biaya = 35 + 85 + 70 = 190.

Jadwal optimal seperti pada Tabel di bawah ini :

| Dari Towing | I&D Station | D&R Station | R&T Station |
|-----------------|-------------|-------------|-------------|
| Mesin | 1 | 3 | 2 |
| Biaya | 35 | 85 | 70 |
| Kumulatif Biaya | 35 | 120 | 190 |

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)

- e) 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)

Jawaban:

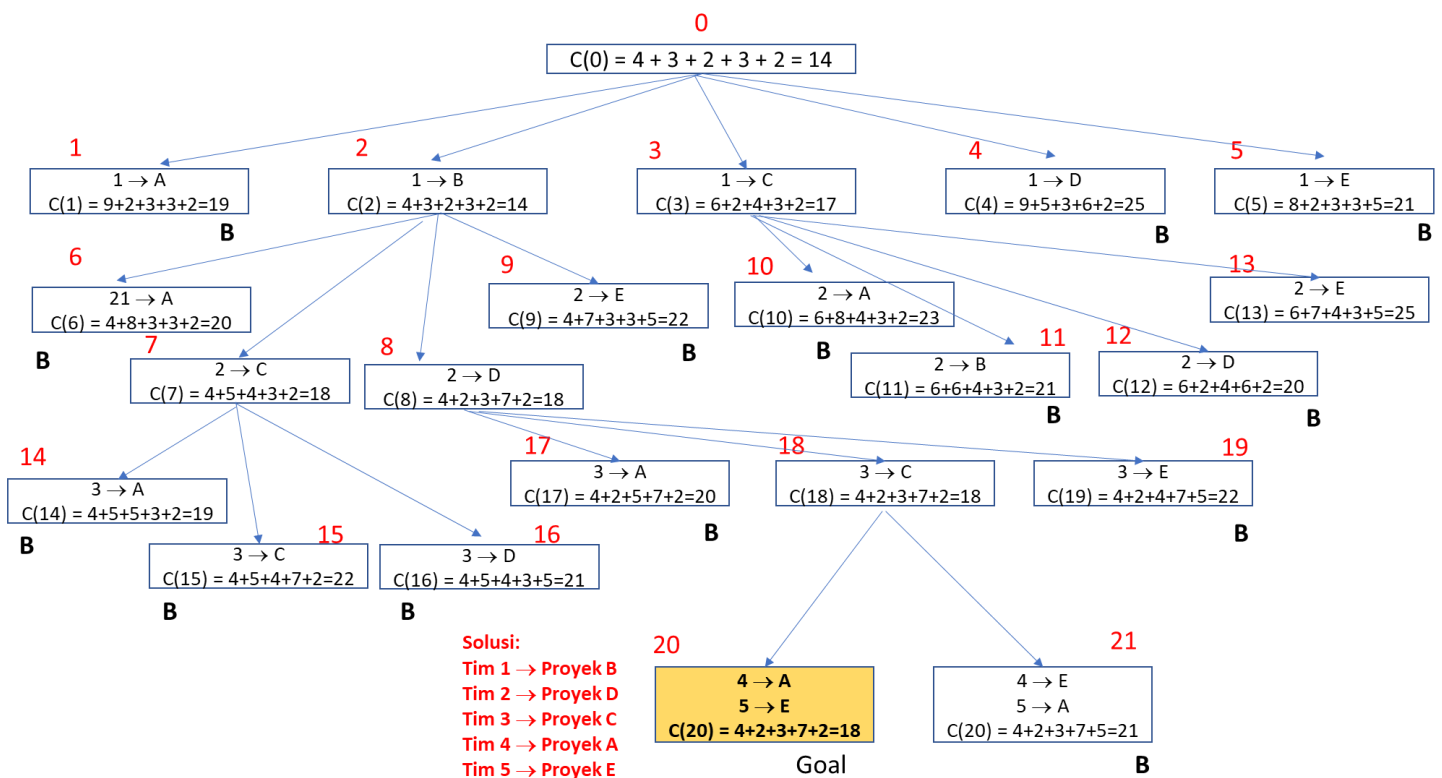
- Salah
- Salah
- Salah
- Salah
- Salah

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, perhatikan 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 |

Jawaban:



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. Perhatikan proses pencocokannya dan jumlah perbandingan karakter pada masing-masing algoritma.
- (b) Diberikan teks sebagai berikut: WELCOMETOMYCOALLISION. Carilah *pattern* COAL dengan algoritma Boyer-Moore. Dalam menjawab soal ini, perhatikan 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)

Jawaban:

(a) Banyak contohnya, antara lain:

Teks: aaaaaabcdef
 Pattern: bcdef

Brute Force

Teks: aaaaaabcdef
 Pattern: ¹bcdef
 ²bcdef
 ³bcdef
 ⁴bcdef
 ⁵bcdef
 ⁶bcdef
 ⁷⁸⁹¹⁰¹¹bcdef

11 kali perbandingan karakter

KMP

Teks: aaaaaabcdef
 Pattern: ¹bcdef
 ²bcdef
 ³bcdef
 ⁴bcdef
 ⁵bcdef

6
bcdef
7891011
bcdef

11 kali perbandingan karakter

(c)

| | | | | | |
|-------|---|---|---|---|---------|
| Huruf | C | O | A | L | lainnya |
| L(i) | 0 | 1 | 2 | 3 | -1 |

| | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| W | E | L | C | O | M | E | T | O | M | Y | C | O | A | L | L | I | S | I | O | n |
| | | | 1 | | | | | | | | | | | | | | | | | |
| c | O | A | L | | | | | | | | | | | | | | | | | |
| | | | | | | 2 | | | | | | | | | | | | | | |
| | | | c | O | A | L | | | | | | | | | | | | | | |
| | | | | | | | | | | 3 | | | | | | | | | | |
| | | | | | | | C | O | A | L | | | | | | | | | | |
| | | | | | | | | | | | 7 | 6 | 5 | 4 | | | | | | |
| | | | | | | | | | | | C | O | A | L | | | | | | |

Ada 7 kali perbandingan karakter

(d) $[A-Za-z]\{2\}\d\{4\}$ artinya string yang diawali oleh dua buah huruf (huruf kapital atau huruf kecil) dan diikuti oleh 4 angka (digit)

String yang bersesuaian: an2021, IM1352, IF2211, ei2023

7. Tentukan prediksi nilai anda untuk kuliah ini (A/AB/B/BC/C/D/E)

(Nilai = 2)