



Program Studi Teknik Elektro ITB

Nama Kuliah (Kode) : Praktikum Pemecahan Masalah dengan C (EL2208)

Tahun / Semester : 2020-2021 / Genap

Modul : 8 - *Algorithms*

Hari, Tanggal Praktikum : Senin, 12 April 2021

Naskah Soal Praktikum

Pembuat Naskah: David Khowanto, Chessa Nur Triejunita

Ketentuan:

1. Kerjakanlah satu dari dua soal berikut pada *template repository* yang anda peroleh ketika mengambil *assignment* di GitHub Classroom praktikum!
2. *Commit* yang dilakukan setelah sesi praktikum berakhir tidak akan dipertimbangkan dalam penilaian.
3. *Header* setiap *file* harus mengikuti format yang telah disediakan pada *file template repository*. *Header* yang tidak mengikuti format tersebut tidak akan dinilai.
4. Buku catatan laboratorium yang berisi *flowchart*, *data flow diagram*, dan analisis kompleksitas waktu dan ruang Big O dari solusi yang anda buat dikumpulkan ke tugas.stei.itb.ac.id paling lambat 24 jam setelah sesi praktikum. Tulis pula alasan anda memilih mengerjakan soal yang anda kerjakan dan mengapa bukan soal yang lainnya!
5. Solusi soal pertama harus dapat dikompilasi dengan perintah `make soal-01` dan menghasilkan *file executable* dengan nama `soal-01`. Demikian pula, soal kedua harus dapat dikompilasi dengan perintah `make soal-02` dan menghasilkan *file executable* dengan nama `soal-02`.
6. Bila diperlukan, sesuaikanlah isi *Makefile* yang tersedia pada *template repository* untuk memenuhi syarat kompilasi dan *file* keluaran di atas!

Soal 1

Gugus bintang (*star cluster*) merupakan suatu kelompok bintang yang terkumpul bersama atau berdekatan dengan jumlah bintang yang sangat besar. Walaupun terpengaruh oleh efek gravitasi, gugus bintang yang terlihat (cahaya tampak) tidak selalu berbentuk bulat, bisa saja oval ataupun memiliki bentuk yang lain akibat dari materi antar bintang (*interstellar matter*) dan masih banyak aspek lainnya.

Ketika tertangkap oleh teleskop, teleskop akan menyimpan informasi tersebut sebagai sebuah citra biner yang hanya terdiri dari angka 1 (terang) dan 0 (gelap). Maka dari itu, buatlah sebuah program yang dapat menghitung jumlah gugus bintang yang ada pada suatu citra berukuran 5x5 piksel. Gugus bintang direpresentasikan oleh angka 1 pada citra biner tersebut. Seperti yang dapat dilihat pada gambar di bawah ini, terdapat total 4 gugus bintang pada citra biner tersebut. Tugas dari program yang dibuat adalah menghitung jumlah gugus bintang pada suatu citra biner.



Gambar 1. Citra biner 5x5 dengan 4 gugus bintang

Catatan:

1. Ukuran citra selalu 5x5 piksel (matriks 2D, 5x5)
2. Citra biner (matriks 2D) hanya terdiri dari angka 1 dan 0
3. Harus menggunakan salah satu algoritma berikut yaitu *Depth First Search* (DFS) atau *Breadth First Search* (BFS). Silakan ditentukan algoritma mana yang lebih baik untuk kasus ini.
4. Tipe data atau struktur data yang digunakan dibebaskan asal menggunakan salah satu algoritma seperti poin nomor 3
5. Hint: Citra biner yang dibentuk ini juga dapat disebut sebagai *Undirected Graph*

Contoh eksekusi program (garis bawah menunjukkan input)

#1

Insert picture:

1 1 0 0 0

0 1 0 0 1

1 0 0 1 1

0 0 0 0 0

1 0 1 1 0

Number of star cluster is: 4

#2

Insert picture:

1 0 0 0 1

0 1 0 1 0

0 0 1 0 0

0 1 0 1 0

1 0 0 0 1

Number of star cluster is: 1

#3

Insert picture:

1 0 0 0 1

0 0 0 0 0

0 0 1 0 0

0 0 0 0 0

1 0 0 0 1

Number of star cluster is: 5

Soal 2

Terdapat input matriks dengan ukuran $n \times m$. Terdapat input *path* yang ingin diperiksa pada matriks apakah *path* tersebut ada atau tidak pada matriks. Buatlah program untuk mengecek apakah *path* terdapat pada matriks atau tidak.

Contoh ilustrasi:

Input matriks:

1	2	3
4	5	6
7	8	9

Path: 1 2 5 8

Output: Path 1-2-5-8 terdapat pada matriks

1	2	3
4	5	6
7	8	9

Keterangan:

- $1 \leq n \leq 10$
- $1 \leq m \leq 10$
- $2 \leq \text{panjang path} \leq n * m$
- Elemen pada matriks dan path ≥ 0
- -1 digunakan untuk batas input *path*
- Gunakan algoritma DFS atau BFS

Contoh eksekusi program (garis bawah menandakan input)

```
#1
Masukkan ukuran matrix: 3 3
Masukkan input matrix:
1 2 3
4 5 6
7 8 9
Masukkan input path yang ingin dicek: 1 2 5 8 -1

Path 1-2-5-8 terdapat pada matriks
```

#2

Masukkan ukuran matrix: 3 3

Masukkan input matrix:

1 2 3

2 3 4

4 6 7

Masukkan input path yang ingin dicek: 2 3 6 -1

Path 2-3-6 terdapat pada matriks

#3

Masukkan ukuran matrix: 2 4

Masukkan input matrix:

5 10 15 20

9 72 91 3

Masukkan input path yang ingin dicek: 3 15 10 72 -1

Path 3-15-10-72 tidak terdapat pada matriks

#4

Masukkan ukuran matrix: 3 8

Masukkan input matrix:

1 6 5 1 5 3 2 4

1 3 2 1 5 0 0 2

2 3 2 2 0 1 0 1

Masukkan input path yang ingin dicek: 2 0 1 0 0 2 4 -1

Path 2-0-1-0-0-2-4 terdapat pada matriks

#5

Masukkan ukuran matrix: 10 10

Masukkan input matrix:

1 8 25 8 6 72 1 9 0 6

27 92 7 9 5 7 7 9 6 25

0 0 82 7 8 66 5 8 2 0

7 3 72 97 4 3 67 1 5 5

1 9 0 6 5 2 9 8 5 1

92 84 7 8 6 11 88 7 5 3

0 0 0 3 2 8 6 9 7 1

7 2 8 27 8 62 72 6 5 5

8 98 72 66 5 7 2 3 1 2

8 9 7 2 6 5 1 0 9 8

Masukkan input path yang ingin dicek: 11 2 3 67 1 5 5 0 2 6 0 9 1 72 -1

Path 11-2-3-67-1-5-5-0-2-6-0-9-1-72 terdapat pada matriks