

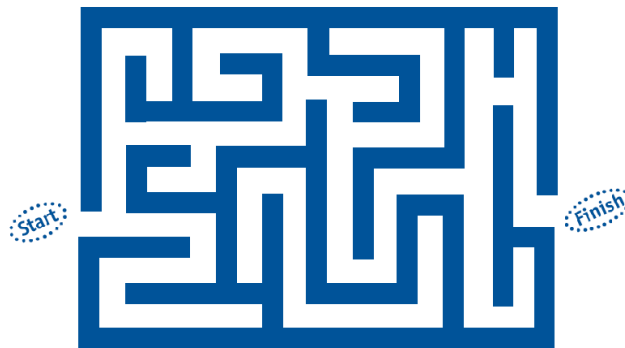
Tugas Kecil 3 IF2211 Strategi Algoritma
Semester 2 tahun 2018/2019

Implementasi Algoritma BFS dan A* pada Persoalan Labirin (*Maze Problem*)

Tugas berkelompok @ 2 orang, boleh lintas kelas
Dikumpulkan: Senin 1 April 2019

Bahasa Pemrograman: C++/Java/Python/C#/Ruby/Perl

Persoalan labirin (*maze problem*) adalah persoalan mencari lintasan dari sebuah titik masuk ke titik keluar di dalam sebuah labirin. Andakan sebuah robot berjalan menyusuri labirin dari sebuah titik masuk ke titik keluar. Robot dapat berjalan dengan menggunakan metode BFS (*Breadth First Search*) atau DFS (*Depth First Search*). Jika robot ingin mencari jalan keluar dengan lintasan terpendek, maka robot dapat menggunakan Algoritma A* (atau A star). Algoritma A* dapat digunakan untuk menentukan lintasan terpendek dari suatu titik ke titik lain. BFS dan DFS termasuk *uninformed search*, sedangkan algoritma A* termasuk ke dalam *informed search*.

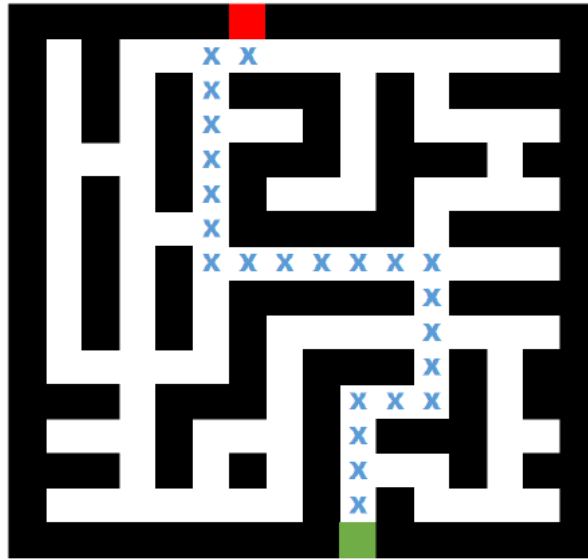


Di dalam Tugas Kecil ini anda diminta membuat masing-masing program pencarian lintasan di dalam labirin dengan menggunakan algoritma BFS (*Breadth First Search*) dan algoritma A*. Pencarian lintasan dengan menggunakan algoritma BFS tidak harus berupa lintasan terpendek, tetapi pencarian lintasan dengan algoritma A* harus menghasilkan lintasan terpendek. Dari hasil pencarian lintasan dengan kedua algoritma maka anda dapat membandingkan lintasan yang dibentuk oleh kedua algoritma. Labirin dinyatakan sebagai sebuah matriks 0/1. Nilai satu menyatakan penghalang (*obstacle*), sedangkan nilai 0 menyatakan grid yang dapat dilalui.

Spesifikasi program:

1. Program menerima input matriks yang merepresentasikan labirin. Matriks yang menyatakan labirin dibaca dari file teks. Input lainnya adalah titik masuk dan titik keluar.
2. Program dapat menampilkan lintasan hasil pencarian solusi dengan algoritma BFS dan A*. Tampilkan lintasan di dalam labirin dengan warna yang berbeda (misalnya merah). Boleh menggunakan pustaka grafika untuk menampilkan gambar labirin (bonus 5) atau cukup tampilan dalam mode teks saja.

Contoh tampilan:



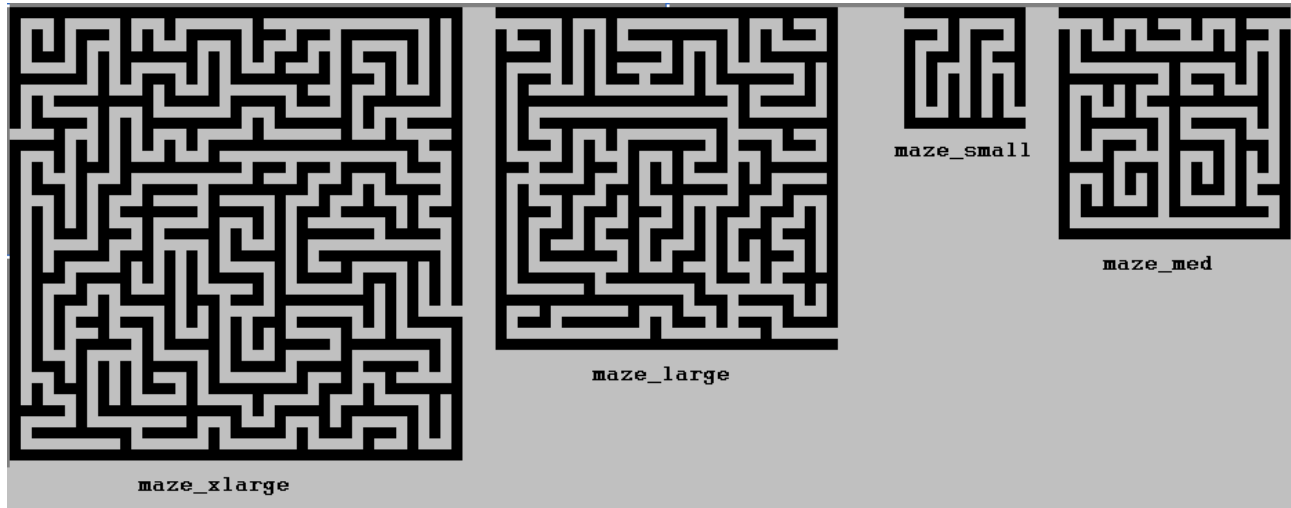
3. Khusus untuk algoritma A*, fungsi *cost* yang digunakan adalah $f(n) = g(n) + h(n)$. Tuliskan di dalam laporan cara menghitung $g(n)$ dan fungsi heuristik $h(n)$.

Berkas yang dikumpulkan: Laporan (*hard copy*) berisi kode program, labirin input, dan *screenshot* yang memperlihatkan lintasan di dalam labirin. Tampilkan hasil untuk lima (5) contoh labirin. Bandingkan hasil keduanya.

Lengkapi tabel berikut untuk memudahkan asisten menilai:

		Centang (✓) jika ya
1	Program dapat menerima input labirin	
2	Program dapat mencari lintasan dengan Algoritma BFS	
3	Program dapat mencari lintasan dengan Algoritma A*	
4	Program dapat menampilkan lintasan di dalam labirin dengan algoritma BFS	
5	Program dapat menampilkan lintasan di dalam labirin dengan algoritma A*	

Lampiran: Contoh 3 labirin sebagai data uji:



Matriksnya:

Maze_small:

```
11111111111111
00001000001
11101011101
10001010001
10111010111
10100010001
10101010101
10101010101
10101010101
10001010100
111111111111
```

Maz_med:

```
11111111111111111111
000010001000100010000
101010101110101011101
101000100000001010001
101111111111111110101
100000000010000010101
111111111010111010101
100010100010001000101
101010101111111111101
101000100010000000101
101111111010111110101
101000001010000010001
101011101011111011101
100010000010001010101
111110111010101010101
100010101010101010001
```

101010101010111010111
 101010001010000010001
 10111111101111111101
 1000000000000000001
 11111111111111111111

Maze_Large:

11111111111111111111111111111111
 0000001000100000000000100010001
 10111010101011111111110101011101
 1000101010100000100010101000001
 1111101010111110101010101111111
 100000101010000010101000000101
 101110101011101110101111110101
 1010000010000000001000100000101
 101111111111111111110101111101
 1010000000000000000000100000001
 101011111111111111111011111111
 1010001000100000000010000010001
 1011101010101111111010111011101
 1000001010001000100010001000001
 1110111011101010101110101111111
 1000000010001010100000101000001
 10101111101110111111110111011111
 1010001000101000110000100010001
 1011101011101110101011101110101
 1010001010001000101010001000101
 1010111010111011101010111011101
 1010100010001010001010101000101
 1010111111101010111010101110101
 1010000000001000100010001000101
 10111111101111111111011101011111
 1000000010000000001000000000101
 11111111111111111101110111110101
 1000100000001000100010000010001
 1011101111111010111010111011111
 10000000000000001000000000100000
 11111111111111111111111111111111

Maze_xlarge:

11111111111111111111111111111111
 1000100000100010000000100000000000101
 1010101110101010111110111011101111110101
 10101010001010001000101000100010000010101
 101110101011111111010101011101010111010101
 10000010101000001010100010001010001010101
 1111111010101110101011111111011101010101
 10000000100010100010000000100010001000101
 10101111111110111110111110111010111111101
 1010000010001000000010001000001010000001
 10111110101011111111101011111110101011111
 00001000101010001000001000000010001000001
 111110101010101010111111111111111110111
 10001010101000100010000000000000000010001
 101010111011111111111011101110111111011101

10100010001000000000001000100010001000001
1011101011101111101111111101110101111101
10001010100010000010000010001000100010001
1010101010111111011111010111011111010111
10101010100010000010001010100000001010001
10101110111010111110101010111111101111101
1010000010001000001010001010000000000001
10101111101110101010111110101111111111101
10101000001010101010000010100010000010101
10111011111010101011111010111110111010101
10100010001000101000001010000000101010101
101011101011111011101110111111111101010101
10101000100000101010000010000000101010000
10101011101110101010101110111110101011111
10101000100010100010101010000010001000001
10101011111010111110101011111011111111101
10100010000010000010100010000010000000101
10111010101011111010111110111110111110101
10001010101010001010000000100010001000001
1010111010101110101111111101110101111111
10100010101000001000001000001000100000101
11111010111111111111101011111011111110101
10000010000000001000100010001010000010101
10111111111011111010111110101110111010101
10000000001000000010000000100000100010001
11

-selamat mengerjakan-