INF1010 – ESTRUTURAS DE DADOS AVANÇADAS TRABALHO G3 – TRAÇAR ROTA

Descrição:

O Trabalho 3 consiste em implementar os algoritmos para calcular o caminho mais curto em um mapa de 40 (lin) x 82 (col). O mapa é uma representação de um desfiladeiro e não há bloqueios. Cada casa (vértice) permite trafegar nas 4 adjacências (norte, sul, leste e oeste; aka cima baixo, direita e esquerda). Também há uma notação especifica (em caracteres) que define o custo do caminho:

- M = 100 (caminho montanhoso necessário escalar)
- R = 5 (rochas, trilha sedimentada, demora um pouco mais, porém é fácil)
- . (ponto) = 1 (livre)

Também há duas marcações especiais: I (letra i) define a partida e F define o destino. Segue representação do mapa (entregue em arquivo).

MMM...R.M..R....R.M..R....R.M..R....R.M..R....R.M..R....R.M..R.M..R.M..R.M..R.M.MM MMM.....R.....MMMMM....R....M...MMMMMM.....R.....M...R.....R.....M....MMM $\mathsf{MMM} \dots \mathsf{R} \dots \mathsf{M} \dots \mathsf{R} \dots \mathsf{MMMMMM} \dots \mathsf{R} \dots \mathsf{M} \dots \mathsf{R} \dots \dots \mathsf{MMM}$

Requisito do Trabalho:

Implementar os algoritmos apresentados como pseudo-código e comparar seus resultados. Os algoritmos A* e Dijkstra são bem parecidos e pode-se usar como base a implementação da busca em largura ou profundidade.

Apresentar os resultados e comparação de:

- Caminho mais curto encontrado (que será o mesmo)
 - o Busca em Largura, Busca em Profundidade, A* e Dijkstra
 - Número de vértices visitados durante o cálculo
- Árvore Geradora Mínima
 - o Prim e Kruskal
- Tempo gasto durante a execução do algoritmo. Segue exemplo:

```
#include <time.h>

clock_t t;

t = clock(); // momento inicial

// executa algoritmo

t = clock() - t; // momento final

printf("Tempo em %f segundos\n", ((double)t)/CLOCKS_PER_SEC);
```

O objetivo final será comparar as diferenças entre esses resultados.

Para ajuda na implementação:

- Consulte a explicação de caminhos mais curtos (slides)
- Se preferir, utilize o ambiente no site https://www.onlinegdb.com/
 - Neste caso, envie o zip do seu projeto para a entrega do trabalho e indique o link para o fork - isso facilita a execução do projeto para correção.

Forma de Avaliação:

Será avaliado se:

- (1) O trabalho atendeu a todos os requisitos especificados anteriormente;
- (2) Os algoritmos foram implementados e aplicados de forma correta;
- (3) O código foi devidamente organizado;
- (4) Os resultados estão corretos;