

Trabalho Prático 1 - 10 pts - Em dupla

## 1 Objetivo

O objetivo deste trabalho é apresentar ao aluno técnicas de implementação de algoritmos de caminhamento em grafos.

## 2 Descrição

Um labirinto é uma construção contendo múltiplas divisões do qual é difícil encontrar a saída uma vez que se está dentro dele. Diversos jogos utilizam labirintos como cenários para seus jogadores, sendo o mais clássico deles o PACMAN. Um jogador pode se mover livremente dentro labirinto, respeitando, claro, as limitações físicas de acesso. Algumas vezes se faz necessário descobrir um caminho de um lugar a outro de forma automática, seja porque o usuário assim o requisitou ou porque a “máquina” precisa de tal rota. Esse cálculo é feito utilizando algoritmos de caminhamento em grafos, como o de amplitude ou de profundidade. Se o caminhamento envolver pesos nas arestas, utiliza-se o algoritmo de Dijkstra.

Neste trabalho, você deve implementar um algoritmo que leia a descrição de um labirinto de um arquivo de entrada, requisiite do usuário dois pontos - um de origem e outro de destino - e faça o cálculo de rota que leve da origem ao destino, se houver tal rota. Após fazer o cálculo de rota, o algoritmo deve imprimir o labirinto na tela, realçando a rota calculada.

O labirinto será descrito por uma matriz de 45 linhas por 55 colunas, da seguinte forma:

- Cada uma das posições do labirinto pode conter uma das 15 peças abaixo:

Núm. Peça	1	2	3	4	5
Peça					
ASCII	16	30	200	17	205

Núm. Peça	6	7	8	9	10
Peça					
ASCII	188	202	31	201	186

Núm. Peça	11	12	13	14	15
Peça					
ASCII	204	187	203	185	206

- O labirinto deve conter uma forma física viável, não sendo possível construções irregulares.

- O arquivo de entrada contendo a descrição do labirinto é um arquivo texto contendo diversas linhas da forma

```
tile <coluna> <linha> <num. da peça>
```

ou seja, cada linha contém a posição de cada peça dentro da matriz de 45x55. Eis um exemplo:

```
tile 0 8 10
tile 14 9 6
tile 15 3 12
tile 16 10 6
tile 9 15 10
tile 3 8 10
tile 12 14 12
tile 2 17 10
tile 16 17 6
tile 3 4 7
tile 14 3 3
```

Após a leitura das peças da matriz, seu programa deve formar um grafo que represente a estrutura do labirinto. Considerando-se que o grafo possui 45x55 posições, isso leva a um grafo com 2475 vértices, cujas arestas dependem da estrutura lida do arquivo, como exemplificado a seguir:

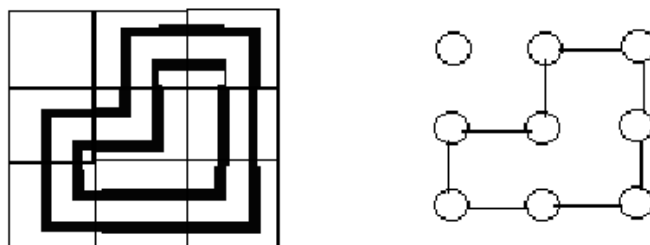


Figura 1: Exemplo do grafo de um labirinto.

Uma vez montado a matriz de adjacência que representa o grafo do labirinto, você pode utilizar qualquer algoritmo de caminhamento para criar uma rota entre os dois pontos lidos. A rota deve ser armazenada em um vetor *Rota*. A seguir, seu programa deve imprimir o labirinto destacando a rota encontrada. Para imprimir cada peça do labirinto, você deve utilizar os caracteres da tabela ASCII dado na tabela acima.

Veja o exemplo a seguir:

Alguns arquivos contendo descrições de diferentes labirintos, assim como um arquivo contendo métodos para imprimir cores diferentes em um terminal de comando DOS podem ser encontrados no learnloop:

Para imprimir diferentes cores utilizando as funções desse arquivo (*ConsoleColor.h*), basta acrescentar a cor antes do texto a ser impresso. Exemplo

```
...
cout << green << "Esse texto é verde.\n"
cout << white << "Esse texto é branco.\n"
...
```

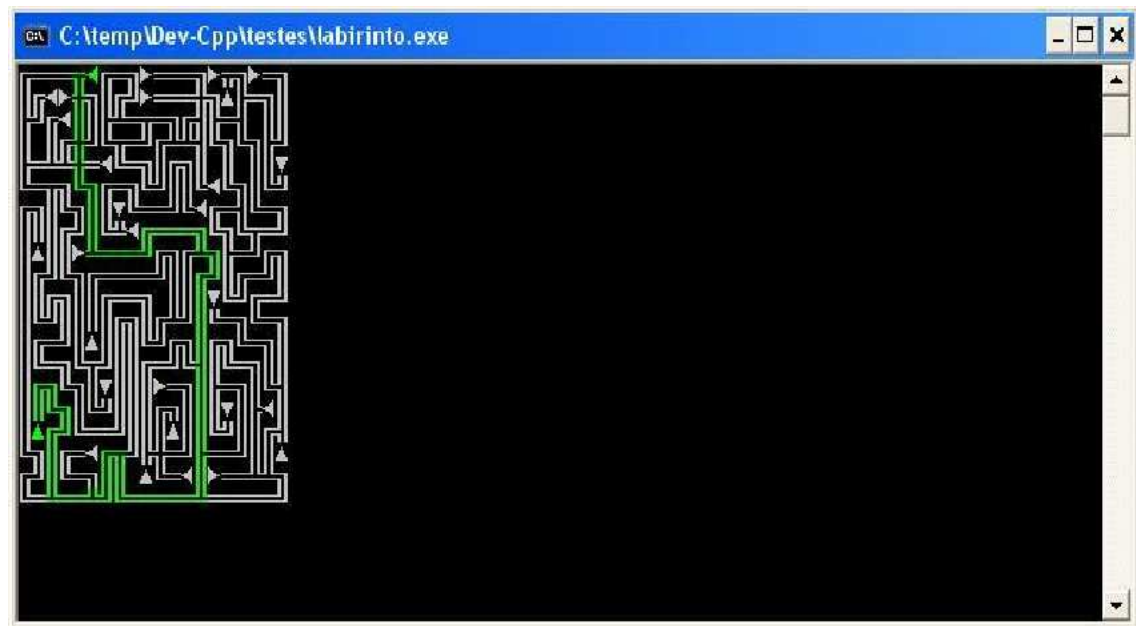


Figura 2: Exemplo de um labirinto.

### 3 Documentação a ser entregue

- Descrição do trabalho
- Descrição da implementação do trabalho: funcionamento do programa, estruturas de dados utilizadas, funções implementadas, escolha do algoritmo
- Manual de funcionamento do programa
- Testes realizados, com as devidas saídas comprovando o funcionamento correto do programa
- Código fonte