

# Introdução à Análise de Algoritmos

2º semestre de 2017 - Turma 04

## Exercício Programa: Labirinto

### 1 Objetivo

O objetivo deste exercício programa é ajudar o explorador a encontrar um caminho em um labirinto, dada uma posição de partida e outra de destino. Havendo mais de um caminho possível, deve ser selecionado aquele considerado melhor a partir de um certo critério escolhido para a execução do programa. O labirinto é representado por uma matriz e a movimentação pelo labirinto pode se dar em 4 direções: para cima, para baixo, para a esquerda e para a direita. Um caminho que leva o explorador da origem ao destino só pode passar por uma posição uma única vez. Além disso, podem haver itens, cada um com um determinado valor e peso, espalhados pelo labirinto. Três são os critérios que devem ser considerados para determinar qual o melhor caminho que leva o explorador da origem ao destino.

1. **Caminho mais curto.**
2. **Caminho que maximize o valor dos itens coletados, sem restrição quanto à carga máxima que o explorador pode levar.**
3. **Caminho que maximize o valor dos itens coletados, considerando que existe um limite de peso máximo que o explorador pode carregar em sua mochila.**

Seu programa deve ler um arquivo de entrada, contendo a definição do labirinto (representado por uma matriz de caracteres), lista de itens espalhados pelo labirinto (coordenadas, valor e peso) e as posições de partida e de destino. Deve ser gerada como saída uma listagem de coordenadas, referentes às posições que fazem parte do caminho encontrado, assim como uma listagem dos itens coletados. Maiores detalhes sobre o arquivo de entrada e a formatação da saída são apresentados a seguir.

### 2 Arquivo de Entrada

Seu programa deve receber dois parâmetros na linha de comando, da seguinte forma:

```
java EP <arquivo com definição do labirinto> <critério>
```

onde o primeiro parâmetro especifica o nome de um arquivo que contém a especificação do labirinto, e o segundo especifica um dos três critérios a ser considerado na procura pelo caminho (um valor inteiro entre 1 e 3). Abaixo um exemplo de como deve ser chamado o EP na linha de comando:

```
java EP mapa1.txt 2
```

O arquivo de entrada deve conter a definição do labirinto, a lista de itens espalhados pelo mesmo, a capacidade máxima de carga da mochila e as posições de partida/destino, respeitando a seguinte formatação:

```
<numero de linhas do labirinto> <numero de colunas do labirinto>
<linha 0>
<linha 1>
<linha 2>
(...)
<número de itens>
<linha item 0> <coluna item 0> <valor item 0> <peso do item 0>
<linha item 1> <coluna item 1> <valor item 1> <peso do item 1>
<linha item 2> <coluna item 2> <valor item 2> <peso do item 2>
(...)
<capacidade da mochila>
<linha posição de partida> <coluna posição de partida>
<linha posição de destino> <coluna posição de destino>
```

A seguir, um exemplo de um arquivo de entrada:

```
7 5
.....
.X.X.
.X.X.
.X.X.
.X.X.
.X.X.
.....
3
4 0 3 3
3 4 7 4
2 0 8 6
5
6 2
0 2
```

A primeira linha do arquivo contém 2 valores inteiros que definem o número de linhas e o número de colunas do labirinto (no caso do exemplo, 7 linhas e 5 colunas). Cada uma das 7 (no caso do exemplo) linhas seguintes contém uma sequência de caracteres de tamanho igual ao número de colunas (5, para o arquivo de exemplo), e cada caractere representa uma posição do labirinto. O caractere “.” indica uma posição livre, enquanto que o caractere “X” representa uma posição bloqueada (ou seja, não pode fazer parte de um caminho). Após a última linha referente ao labirinto, há uma nova linha com um valor inteiro que define a quantidade de itens espalhados pelo labirinto (3 para o exemplo). E em seguida uma linha referente a cada item. Cada linha referente a um item possui 4 valores inteiros: linha e coluna do item no labirinto, seguidos do valor e do peso do item. Finalmente, as 3 últimas linhas, também com valores inteiros, definem os atributos referentes ao explorador. A antepenúltima

linha define a capacidade máxima de carga da mochila (5, no caso do exemplo), a penúltima linha as coordenadas (linha e coluna, respectivamente) da posição de partida (linha 6 e coluna 2, para o exemplo), e a última linha as coordenadas da posição de destino (linha 0 e coluna 2, para o exemplo).

### 3 Saída

A saída gerada pelo seu programa deve ser impressa na saída padrão e deve respeitar a seguinte formatação:

```
<tamanho do caminho encontrado>
<linha da posição 0> <coluna da posição 0>
<linha da posição 1> <coluna da posição 1>
<linha da posição 2> <coluna da posição 2>
(...)
<quantidade itens coletados> <valor total itens coletados> <peso total itens coletados>
<linha item coletado 0> <coluna item coletado 0>
<linha item coletado 1> <coluna item coletado 1>
<linha item coletado 2> <coluna item coletado 2>
(...)
```

Para o arquivo de entrada apresentado como exemplo, a saída gerada deve ser a seguinte quando a opção de critério 1 é utilizada:

```
7
6 2
5 2
4 2
3 2
2 2
1 2
0 2
0 0 0
```

Já para a opção 2, a saída deve ser a seguinte:

```
11
6 2
6 1
6 0
5 0
4 0
3 0
2 0
1 0
0 0
0 1
```

```
0 2
2 11 9
4 0
2 0
```

E finalmente, para a opção 3:

```
11
6 2
6 3
6 4
5 4
4 4
3 4
2 4
1 4
0 4
0 3
0 2
1 7 4
3 4
```

Para entender melhor o funcionamento da opção 3, suponha que a capacidade de carga de mochila seja 8, ao invés de 5. Neste caso a saída gerada pelo programa deve ser:

```
11
6 2
6 1
6 0
5 0
4 0
3 0
2 0
1 0
0 0
0 1
0 2
1 8 6
2 0
```

pois apesar de o caminho passar pelo item que tem valor 3/3 (valor = 3 e peso = 3), não é viável guardá-lo na mochila. Por outro lado, se a coleta do item 3/3 fosse feita em detrimento do item 8/6, o valor somado dos itens coletados não seria o maior possível.

## 4 Prazos e Entrega

Este EP deve ser realizado **individualmente**, implementado na linguagem **Java**, e sua entrega deve ser feita até o dia 15/12/2017 às 23:55 pelo TIDIA-Ae. Para entregá-lo, crie uma pasta nomeada com seu número USP e coloque o(s) fonte(s) de seu EP dentro dela. Em seguida compacte a pasta no formato **ZIP** (criando um arquivo também nomeado com seu número USP mais a extensão “.zip”) e envie este arquivo pelo TIDIA-Ae.

### Observações importantes:

- Não se esqueça colocar, no início de cada arquivo fonte criado, um comentário contendo seu nome e número USP.
- Seu EP deve seguir à risca as especificações de entrada e saída deste enunciado. A não conformidade com as especificações irá acarretar em descontos na nota.