

Questão 4 - Aventureiro

- Identificar a melhor estrutura de dados para representar o mapa e a implemente;

Para representar o mapa utilizamos uma estrutura de matriz bidimensional (linha X coluna). Essa estrutura apresenta torna simples a implementação de funcionalidades como encontrar o ponto de partida e o ponto de chegada. Também facilita operações como marcar o caminho e imprimir o próprio mapa.

- Identificar e implementar o algoritmo para percorrer o mapa e atingir o destino.

Para encontrar o caminho destino, primeiramente geramos um grafo a partir do mapa armazenado em matriz. Em cada nó do grafo montamos uma lista de vizinhos que sejam “Caminho Livre”, “Entrada” ou “Destino”. Dessa forma, temos os possíveis caminhos representados em um grafo, onde é possível efetuar mecanismos de busca para encontrar o destino a partir da entrada.

- Permitir que o caminho possa ser realizado de volta, sem que o aventureiro tenha que refazer caminhos inúteis. Ainda, o caminho de volta não pode ser um caminho novo. Ele precisa voltar por onde ele veio, usando a rota mais eficiente.

Para percorrer o caminho fizemos a busca no grafo utilizando o algoritmo de busca em largura (*breadth-first search*), ou *busca BFS*. Este algoritmo utiliza estruturas de filas e conjuntos para armazenar o caminho que está sendo feito. Esse algoritmo garante o menor caminho em relação ao número de passos.

Análise de Complexidade, por etapas.

1. Leitura do mapa.

$O(nm)$, onde n é o número de linhas e m é o número de colunas do mapa.

2. Geração do grafo de caminhos

$O(nm)$ no pior caso (quando todas as células são caminháveis), já que o grafo armazena todas as posições possíveis como nós com vizinhos.

3. Busca do destino em BFS.

$O(V + E)$, onde:

- V = número de vértices = $O(nm)$ no pior caso.
- E = número de arestas = até $4 \times V$, portanto $E = O(nm)$.
- Resultado: $O(nm)$.

4. As marcações de caminho e impressão de mapa percorrem exatamente 1 posição do mapa. Resultando $O(nm)$

Complexidade final: $O(nm)$

Podemos concluir que a complexidade do algoritmo é $O(nm)$, sendo n o número de linhas do mapa e m o número de colunas.