

Segundo trabalho Prático

1- Introdução

O objetivo deste trabalho é aplicar o conceito de pilhas e listas para que um robô fictício percorra um labirinto a partir de um portão de entrada até um portão de saída.

Cada labirinto será representado por uma matriz, a ser lida de um arquivo, com l linhas e c colunas. Seja M tal matriz. Então M irá conter apenas dois tipos de entradas, a saber os caracteres $\#$ e 0 . Toda entrada $\#$ representa um obstáculo intransponível, isto é, uma parte dos muros que formam o labirinto. Toda 0 representa uma área livre, isto é, um lugar por onde o robô pode andar. Após ter encontrado a saída, você deverá imprimi-la em um arquivo de saída. Para cada labirinto dado, é possível que exista mais de uma saída. A rota de fuga que você deve encontrar e imprimir é a rota de fuga mais curta, isto é, a que requer o menor número possível de passos. As paredes de qualquer labirinto dado serão conexas, isto é, com exceção dos portões de entrada e saída, de posse de um lápis e de uma folha de papel, é sempre possível desenhar a planta do labirinto sem tirar o lápis do papel. O arquivo de entrada terá na sua primeira linha os números l e c nesta ordem separados por espaço em branco. Na segunda linha, terá quatro inteiros separados por espaços em branco, a saber x_e , y_e , x_s e y_s , que são os índices das entradas da matriz que representam, respectivamente, o portão de entrada e saída do labirinto. Nas l seguintes linhas desse arquivo, estará escrita a matriz (o labirinto) conforme especificado acima.

2- Restrições

1. o código deve ser feito em usando a linguagem C;
2. implementar os TAD's pilha e Lista;
4. utilizar um vetor de tamanho fixo para armazenar os dados desperdiça memória quando poucos dados são utilizados. Logo, implemente as estruturas usando alocação encadeada;
5. não é permitido o uso de chamadas recursivas de funções;
6. a documentação de seu TP deve conter no máximo 8 páginas;
7. o código deve compilar usando C padrão (ANSI C). Basta evitar utilizar funções específicas de uma determinada plataforma(Windows, Linux, etc).
7. código deve compilar sem nenhum warning! Você pode verificar se seu código compila sem warnings com o seguinte comando no Linux (opcional):
`gcc -Werror arquivos.c`

3- Entrada e Saída

O seu código deve receber o tamanho o nome do arquivo que representa o labirinto. Deve ser exibido na tela o(s) caminho(s) mais curto(s).

4- Documentação

Escreva um documento explicando o seu código e avaliando o desempenho de sua implementação. Separe em cinco seções:

introdução, implementação, Estudo de Complexidade, conclusão e referências. Sua documentação deve conter no máximo 8 páginas.

1 – Introdução

Escreva uma pequena introdução, dando uma ideia geral do sistema e de seu funcionamento.

2 – Implementação

Descrição sobre a implementação do programa. Deve ser detalhada as estruturas de dados utilizadas, o funcionamento das principais funções utilizadas, o formato de entrada e saída de dados, bem como decisões tomadas relativas aos casos e detalhes de especificação que porventura estejam omissos no enunciado.

3 – Estudo de Complexidade

Análise de complexidade das funções implementadas no sistema. Notação O.

4 – Conclusão

Resuma o que você fez e deixou de fazer neste TP. Explique quais foram as dificuldades que você encontrou para fazer este TP.

5 – Referências

Cite as fontes utilizadas para realização deste TP, mesmo que sejam sites da internet.

Observações:

- a) Data de Entrega: 04/11/2019;
- b) O é individual;
- c) Valor: 5,0 pontos
- d) Comecem logo, pois a data de entrega jamais estará tão distante como agora!!!