Instituo Federal sudeste de Minas Gerais – Campus Barbacena
Curso de Tecnologia em Sistemas para Internet
Disciplina: Estruturas de Dados I
Prof. Wender Magno Cota

Terceiro Trabalho Prático

1- Introdução

O objetivo deste trabalho é aplicar o conceito de pilhas e listas para que um robô percorra um labirinto a partir de um portão de entrada até um portão de saída. Os movimentos permitidos são: **esquerda**, **direita**, **cima** e **baixo**(nessa ordem).

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 encontrar a saída, você deverá imprimi-la. 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 xe , ye , xs e ys, que são os índices das entradas da matriz que representam, respectivamente, o portão de entrada e saída do labirinto. Nas L linhas desse arquivo, estará escrita a matriz (o labirinto) conforme especificado acima.

Exemplo de um Labirinto(13x40)

13 40

```
########################################
E0000000000000000000000000#000#000000#
#0#0#0#0#0##0#0#0######0#0#000#0#0#0##0#
#0#0#####0##0#0#000000#0#0#0#0000000000S
#0#000#0#0####0#0####0000000000#0#0###0#
#0######00000##0#000000####0##0#0#0#0#0#
#0############0#0##########0##0#0###0#0#
#0#000000000000000000000000#000000000000#
#0#000#0#0#0#0#0#0#0#0#0#0#0###0#0#0##0#
#00000#000#0000#0000#00000#0#0#000#0##0#
#0#0#########000000000000##000###0####0#
#0000000000000#0000000#00000#0000000000#
########################################
```

Onde E representa a posição de entrada e S a de saída.

2- Restrições

1. o código deve ser feito em usando a linguagem C;
2. implementar os TAD's pilha e Lista;
3. utilizar um vetor de tamanho fixo para armazenar os dados desperdiça memória quando poucos dados são utilizados. Logo, implemente as estruturas usando alocação encadeada;
4. não é permitido o uso de chamadas recursivas de funções;
5. a documentação de seu TP deve conter no máximo 8 páginas;
6. o código deve compilar usando C padrão (ANSI C). Basta evitar utilizar funções específicas de uma determinada platafoma(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 programa deve receber, através da linha de comando, o nome do arquivo que representa o labirinto e o arquivo de saída(com a rota de fuga).

4- Documentação

Escreva um documento explicando o seu código e avaliando o desempenho de sua implementação. Separe em cinco seções:
introdução, implementação, experimento, conclusão e referências. Sua documentação deve conter no máximo 10 páginas.

Introdução

Defina o problema com suas próprias palavras.

Implementação

Separe entre a implementação da pilha, lista e do algoritmo para resolver a rota de fuga. Descreva o que cada função que você implementou faz, sua complexidade, o que ela recebe e o que retorna. Descreva os passos (pseudo algoritmo) de sua abordagem.

Experimentos

Faça cada um dos itens abaixo.

1. Execute o seu código para tamanhos de labirintos :10x10, 20x20, 30x30 ,13x40 e 40x40.
2. Para cada tamanho de labirinto, execute 3 vezes e calcule o tempo médio para execução.
3. Faça um gráfico do número de posições posições de cada labirinto (eixo horizontal) pelo tempo para execução (eixo vertical).

Conclusões

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

Referências

Cite as fontes pesquisadas

Pontuação Extra

Para cada labirinto, é possível que exista mais de um caminho que leva da entrada até a saída. O trabalho receberá pontuação extra nas situações:
a) Mostrar todos os caminhos possíveis.(0,75 ponto extra)

b) Calcular o(s) caminho(s) mais curto(s).(0,75 ponto extra)

Observações:

1. Data de Entrega: 24/03/2020.
2. O trabalho é individual.
3. Valor: 3,0 potos.
4. Trabalhos com cópias integrais ou parciais receberão nota zero.
5. Usem comentários na dose certa.
6. O programa deve ser devidamente modularizado, evitando repetição de código.
7. Utilizar comentários (na dose certa).
8. Comecem logo, pois a data de entrega jamais estará tão distante como nesse momento.