

UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
TEORIA DOS GRAFOS
INF-09348 e PINF-6037

1º Trabalho computacional – 2016-2 – Profa Claudia Boeres

O primeiro trabalho computacional será a implementação de três algoritmos estudados na disciplina para geração e solução de labirintos perfeitos. Um labirinto perfeito pode ser definido como um grafo conexo acíclico (uma árvore), isto é, só há um caminho que interliga quaisquer par de vértices do grafo. Além disso, em nossa aplicação, há a necessidade de dois vértices especiais, a entrada (C) e a saída (N') do labirinto, como pode ser visto na figura figura 1.

A entrada do algoritmo será uma malha quadrada de lado n (vértices).

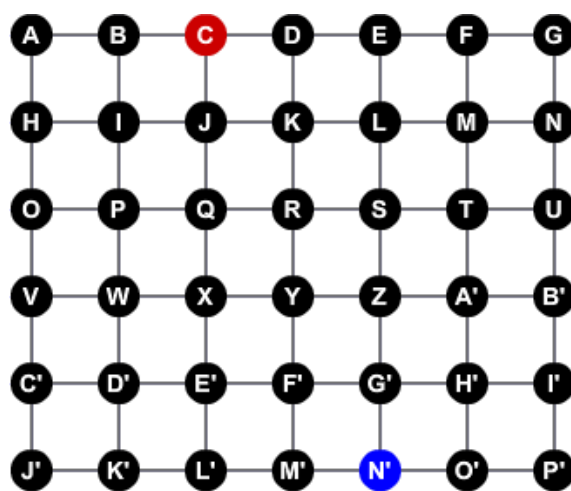


Figura 1: Exemplo de entrada

Essa malha será processada a partir do vértice correspondente à entrada do labirinto por um dos algoritmos listados adiante. Ao final do processamento devemos ter um grafo conexo acíclico que representa o labirinto, figura 2.

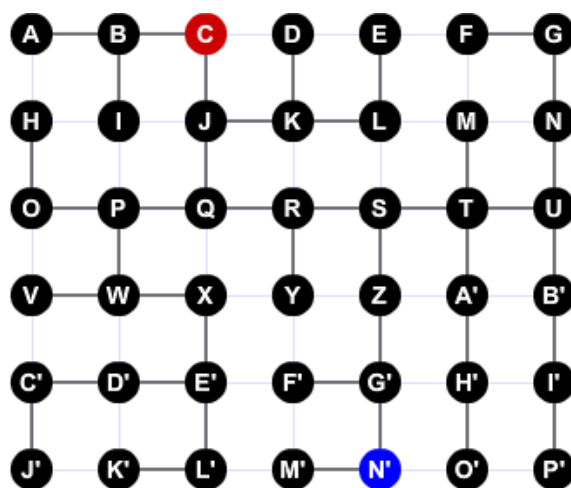


Figura 2: Exemplo de saída

Além do labirinto, o trabalho também exige que a solução (gabarito) daquele labirinto seja apresentada na forma de um caminho, isto é, a lista de vértices que

devem ser atravessados partindo da entrada do labirinto e chegando na saída. Se o labirinto for gerado pelos algoritmos de Prim ou Kruskal esse processamento deve ser realizando usando o algoritmo de busca em largura.

Algoritmos

Cada dupla terá que implementar o algoritmo de busca em profundidade e um algoritmo (a escolha) de árvore geradora mínima, ambos para geração do labirinto. Além disso, o algoritmo de busca em largura deve ser implementado para solucionar os labirintos gerados pelos algoritmos de Prim e Kruskal.

Para os labirintos gerados pela busca em profundidade a solução é trivial, pois ao final da geração do labirinto a pilha conterá a solução e, portanto, não há a necessidade de se aplicar nenhum método de solução. Ao invés disso simplesmente pode-se imprimir os vértices que restaram na pilha.

Aleatoriedade

Sempre que um vértice tiver múltiplas opções de qual aresta selecionar, essa escolha deve ser feita usando as funções pseudo-aleatórias da linguagem escolhida. Caso contrário, se mantiver fixo o tamanho da malha, o método sempre irá produzir a mesma saída. Por isso, é importante que os métodos implementados sorteiem entre as possibilidades.

Execução

Deve-se executar e anotar os tempos de cada etapa, gerar e resolver, para cada algoritmo com instâncias de diferentes tamanhos n , onde n representa quantos vértices o lado da malha possui. Os valores para n são: 10, 50, 100, 500, 1000. Para cada valor de n um mínimo de 10 execuções são necessárias e, junto com os resultados individuais, mostrar a média dos tempos.

Seminários

Cada grupo deve fazer uma apresentação de no máximo 15 minutos contendo uma breve explicação dos algoritmos implementados e aplicados ao labirinto, detalhes da implementação, estruturas de dados utilizadas, apresentação e análise dos resultados obtidos.

Entrega

O material deve ser enviado para o Diego (e-mail: diego.lucchi@gmail.com, com cópia para boeres@inf.ufes.br), no mesmo dia da apresentação do seminário. O material consiste em:

- arquivo dos slides apresentados
- compactado dos códigos implementados
- arquivo compactado dos grafos de entrada gerados

Data dos seminários e entrega do material: 26/10/2016

Fontes adicionais

- https://en.wikipedia.org/wiki/Maze_generation_algorithm
- https://en.wikipedia.org/wiki/Maze_solving_algorithm
- https://rosettacode.org/wiki/Maze_generation
- <http://datagenetics.com/blog/november22015/index.html>