TRABALHO DE PROGRAMAÇÃO – ALGORITMOS EM GRAFOS – 2017/1

ENUNCIADO VÁLIDO PARA TODOS OS TRABALHOS:

Considere n pontos no plano Cartesiano, onde as coordenadas de cada ponto (x, y) são números reais satisfazendo $0 \leq x \leq 1$ e $0 \leq y \leq 1$. A partir deste conjunto de pontos definimos implicitamente um grafo G, onde cada vértice é um ponto (x, y) e existe aresta entre quaisquer dois pontos, cujo peso é exatamente a distância entre os dois extremos da aresta. Lembre que a distância entre os pontos (x, y) e (z, w) é dada por:

$$sqrt((x - z)^2 + (y - w)^2).$$

Construa quatro grafos G_{50}, G_{100}, G_{500} e G_{1000}, onde cada grafo G_n é obtido criando n pontos (x, y) aleatoriamente. Lembre que x e y são números reais satisfazendo $0 \leq x \leq 1$ e $0 \leq y \leq 1$.

TRABALHO 1

Implemente (em qualquer linguagem) o algoritmo de **Busca em Profundidade** e execute-o sobre os quatro grafos construídos. Em cada execução, você deve escolher aleatoriamente a raiz da busca. O resultado de cada execução deve ser exibido da seguinte forma: imprima inicialmente os n pontos do grafo de entrada, e a seguir acrescente as arestas da árvore de profundidade, uma por uma, para que se veja a construção gradativa da árvore. Obs: escolha uma velocidade adequada de impressão das arestas, para uma boa visualização.

Repita as quatro execuções, agora considerando que cada grafo G_n é direcionado da seguinte forma: a direção da aresta ligando dois pontos (x, y) e (z, w) é de (x, y) para (z, w) se $x < z$, e de (z, w) para (x, y) caso contrário. A raiz da busca deve ser o ponto (x, y) com o valor mínimo de x.

OBSERVAÇÃO: PARA MELHOR VISUALIZAÇÃO DA ÁRVORE, CONSIDERE QUE OS VIZINHOS DE CADA VÉRTICE SÃO APENAS OS K PONTOS MAIS PRÓXIMOS, ONDE K = parte inteira de log n

TRABALHO 2

Implemente (em qualquer linguagem) o algoritmo de **Busca em Largura** e execute-o sobre os quatro grafos construídos. Em cada execução, você deve escolher aleatoriamente a raiz da busca. O resultado de cada

execução deve ser exibido da seguinte forma: imprima inicialmente os *n* pontos do grafo de entrada, e a seguir acrescente as arestas da árvore de largura, uma por uma, para que se veja a construção gradativa da árvore. Obs: escolha uma velocidade adequada de impressão das arestas, para uma boa visualização. Coloque cores nas arestas da seguinte forma: arestas do nível 0 para o nível 1 recebem uma cor, do nível 1 para o nível 2 outra cor, e assim por diante, usando sempre cores distintas.

OBSERVAÇÃO: PARA MELHOR VISUALIZAÇÃO DA ÁRVORE, CONSIDERE QUE OS VIZINHOS DE CADA VÉRTICE SÃO APENAS OS K PONTOS MAIS PRÓXIMOS, ONDE K = parte inteira de log n

TRABALHO 3

Implemente (em qualquer linguagem) o **Algoritmo de Dijkstra** e execute-o sobre os quatro grafos construídos. Em cada execução, você deve escolher aleatoriamente a raiz da árvore de Dijkstra. O resultado de cada execução deve ser exibido da seguinte forma: imprima inicialmente os *n* pontos do grafo de entrada, e a seguir acrescente as arestas da árvore de Dijkstra, uma por uma, para que se veja a construção gradativa da árvore. Obs: escolha uma velocidade adequada de impressão das arestas, para uma boa visualização.

TRABALHO 4

Implemente (em qualquer linguagem) o **Algoritmo de Kruskal** e execute-o sobre os quatro grafos construídos. O resultado de cada execução deve ser exibido da seguinte forma: imprima inicialmente os *n* pontos do grafo de entrada, e a seguir acrescente as arestas da árvore geradora mínima, uma por uma, para que se veja a construção gradativa da árvore. Obs: escolha uma velocidade adequada de impressão das arestas, para uma boa visualização.

TRABALHO 5

Implemente (em qualquer linguagem) o **Algoritmo de Prim** e execute-o sobre os quatro grafos construídos. Em cada execução, você deve escolher aleatoriamente a raiz da árvore geradora a ser construída. O resultado de cada execução deve ser exibido da seguinte forma: imprima

inicialmente os *n* pontos do grafo de entrada, e a seguir acrescente as arestas da árvore geradora mínima, uma por uma, para que se veja a construção gradativa da árvore. Obs: escolha uma velocidade adequada de impressão das arestas, para uma boa visualização.