Pedro Scarpioni - 31075

1 -

Nos dois grafos, os dois algoritmos diferentes irão gerar duas árvores geradoras mínimas e custos mínimos semelhantes. Dessa maneira, entendemos que a diferença que fará valer o estudo de qual algoritmo deve ser utilizado está na complexidade de seu código. Neste caso, os dois algoritmos podem ser utilizados em qualquer um dos grafos. As tabelas de execução são as seguintes:













2 -

Lógica do algoritmo:

vetPred (vPred, s)

1. x = s
2. InicializaFila(s)
3. enquanto fila != vazia faça
4. para cada y em vPred faça
5. se x == y faça
6. InsereFila(y)
7. // Insere na posição k (inicia com 0) a aresta de ligacao
8. VetorPred = InsereVetorPred(x+”->”+y, k)
9. k = k+1
10. RemoveFila()
11. Retorna VetorPred

* Inicializa X com o vértice ‘s’.
* Busca o vértice X no vetor de predecessores.
  + Caso encontre um vértice Y que é precedido por X, este vértice é ‘escolhido’...
  + O vértice Y é armazenado em uma fila.
  + Na fila de predecessores é armazenado X-Y
* Após percorrer todo o vetor de predecessor, retira-se o X da fila.
* Repete-se este processo até percorrer todo o algoritmo.

3 –

Neste grafo temos que, durante a execução de Prim (considerando que iniciamos no vértice a):

* Ao ter explorado os vértices ‘a’, ‘b’ e ‘c’, o algoritmo terá duas opções para seguir a aquisição da árvore geradora mínima.
  + Explorar o vértice d (peso 4).
  + ou Explorar o vértice e (peso 4).

Já durante a execução do algoritmo de Kruskal, teremos que:

* Após explorar as arestas “ab”, “bc”, “de” e “fg” teremos duas opções para montar duas árvores geradoras diferentes, sendo elas:
  + Explorar a aresta “ad”, “dg” ou “be” (todas peso 4). Cada uma delas irá gerar árvores diferentes.

4 –

Para resolver o problema proposto, realizamos uma modelagem da seguinte maneira:

* Utilizou-se o algoritmo de Pruskal;
* Para a Letra A conectamos um caminho à olho nu (sem nenhum algoritmo) entre A e J;
* Na Letra B conectamos os vértices que faltavam consultando o algoritmo de Pruskal;
* Na Letra C conectamos TODOS os vértices com o menor custo possível através do algoritmo de Pruskal. Em seguida o comparamos com a soma dos valores obtidos na letra A e B;
* Para realizar os passos acima, montamos uma tabela que continha as arestas e os seus respectivos pesos (em ordem crescente de tamanho):



* + Letra A)

Para conectar A até J, utilizamos as seguintes arestas:

AD – DH – HJ

Com isto, tivemos um custo de $21.000;

* + Letra B)

Para conectar os vértices restantes, utilizamos as arestas:

DG, EG, AC, EI, FH, AB.

Com isto, tivemos um custo de $29.500;

* + Letra C)

Para conectar todos os vértices através do algoritmo de Pruskal, utilizamos os vértices:

DG, EG, AC, EI, FH, IJ, AB

Com isto, temos um gasto de $35.000;

Ou seja, com isto temos algumas considerações:

* Para conectar a energia de A até J o custo é menor do que conectar TUDO;
* Entretanto, a longo prazo, é mais satisfatório realizar todas as conexões de uma vez ($35.000 agora do que $50500 depois);