2ª Lista de Algoritmos e Grafos – 2022/1
Prof. Vinícius Gusmão

1) Diga se a afirmativa é verdadeira ou falsa, apresentando justificativa.

   a) O algoritmo de Dijkstra para caminhos mínimos a partir de origem fixa funciona para todos os digrafos que não possuem ciclos de custo negativo.

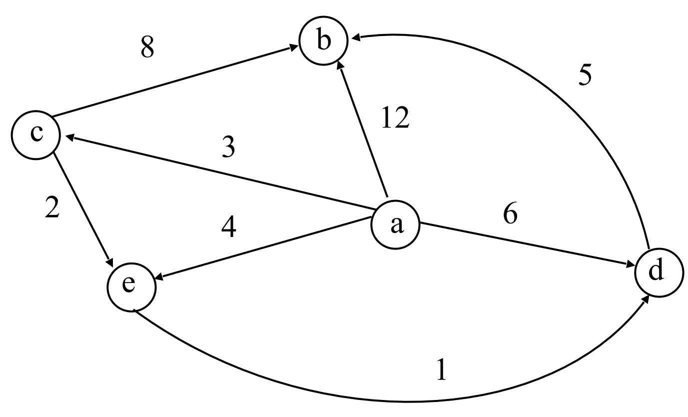
   b) É possível resolver o problema da árvore geradora *máxima* para um certo grafo *G* rodando o algoritmo de Kruskal para árvore geradora mínima num grafo *G'* que tenha os mesmos vértices e arestas de *G*, mas onde o custo de cada aresta seja igual ao da aresta correspondente em *G* multiplicado por -1.

   c) A técnica da programação dinâmica por memoização permite resolver problemas numa abordagem *top-down*, reduzindo sempre, em relação à abordagem tradicional de programação dinâmica *bottom-up*, a quantidade de soluções de subproblemas que precisam ser efetivamente computadas.

   d) O laço do algoritmo seguinte, que descobre se um inteiro $x > 1$ é primo ou composto, é executado no máximo x-2 vezes, cada uma das quais demandando tempo constante. Portanto, trata-se de um algoritmo que roda em tempo polinomial no tamanho da entrada.

      para todo *j* de 2 a *x*-1
          se *x* for divisível por *j* retorne 'composto'
      retorne 'primo'

2) Seja o digrafo *D* abaixo. Deseja-se determinar os caminhos de custo mínimo entre o vértice *a* (origem fixa) e os demais vértices de *D*.



Suponha que o algoritmo dado a seguir será executado.

```
Entrada: custo[1..n,1..n] // matriz de adjacências
para cada vértice v de D: d[v] = ∞; ant[v] = null;
d[a] = 0; ant[a] = null;
seja L uma ordenação topológica para os vértices de D;
para cada vértice v, na ordem em que aparecem em L:
     para cada vértice w que é vizinho de saída de v:
          relaxar(v,w);
para cada vértice v de D: imprima v, d[v], ant[v];
```

a) Escreva em pseudocódigo a função *relaxar*, cuja entrada compreende os vértices incidentes à aresta que deve ser relaxada, bem como os vetores *d* e *ant* a serem (possivelmente) atualizados.
b) Exatamente quantas chamadas a *relaxar* são feitas ao longo de sua execução quando a entrada é o digrafo *D* dado?
c) O algoritmo acima funciona para todo digrafo acíclico e sem arestas negativas?
d) Suponha que, ao invés do algoritmo acima, o algoritmo de Bellman-Ford seja executado para esta entrada *D*. Quantas chamadas a *relaxar* são feitas ao longo de sua execução (desconsiderando a etapa final do algoritmo, que verifica a existência de ciclos de custo negativo)?

3) Simule a execução das 2 (duas) primeiras iterações do laço principal do algoritmo de programação dinâmica de Floyd-Warshall para caminho mínimo entre todos os pares de vértices, quando a entrada é o digrafo *D* da questão 2. Suponha os vértices considerados em ordem alfabética no laço principal do algoritmo.
   a) Ao fim da segunda iteração, quais os antecessores *ant*(*i,j*) e estimativas *d*[*i,j*] para o custo de um caminho mínimo entre cada par de vértices *i,j* de *D*?
   b) Escreva, em pseudocódigo, como é feito o teste e a eventual atualização de uma célula *d*[*i,j*] e *ant*[*i,j*], a cada iteração, em função dos valores anteriores guardados nas matrizes de estimativas e de antecessores.

4) Seja *G* o grafo subjacente ao mesmo digrafo *D* da questão 2 (isto é, *G* é idêntico a *D*, exceto pelo fato de que suas arestas não são orientadas).
   a) Simule a execução das 3 (três) primeiras iterações do algoritmo guloso de Kruskal para determinação de uma árvore geradora de custo mínimo com entrada *G*, apresentando a estrutura obtida ao fim de sua terceira iteração.
   b) Idem, trocando Kruskal por Prim, e considerando *c* como o vértice inicial.

5) Seja mais uma vez o digrafo *D* da questão 2, onde os pesos das arestas representam suas capacidades, no contexto do problema do fluxo máximo. Seja *a* a fonte e *c* o sumidouro da rede. Suponha que o método de Ford-Fulkerson esteja sendo aplicado

e que, num dado momento, o valor do fluxo na rede é igual a 7 unidades de fluxo, das quais 5 unidades estão indo da fonte ao sumidouro pelo caminho *a-c*, e 2 unidades estão indo pelo caminho *a-b-c*.

a) Desenhe o que seria, nesse momento, a rede residual.
b) Aponte um caminho aumentante (qualquer) e indique qual seria seu efeito na atualização do fluxo atual na rede pelo algoritmo de Ford-Fulkerson.

6) Preciso ir de bicicleta da cidade A à cidade B, cidades estas ligadas por uma única estrada. Carrego na bicicleta um *bidon* (garrafinha) de 600 mililitros, e preciso ingerir 100 mililitros de água a cada 5 Km pedalados, do contrário desidrato e morro. Carrego comigo um mapa que indica a localização exata dos postos de combustível ao longo daquela estrada, locais esses onde posso beber água e encher minha garrafinha.

   a) Dê um algoritmo guloso que garanta uma viagem com número mínimo de paradas para abastecimento.

   b) Prove a corretude de seu algoritmo usando argumento do tipo *cut and paste*.