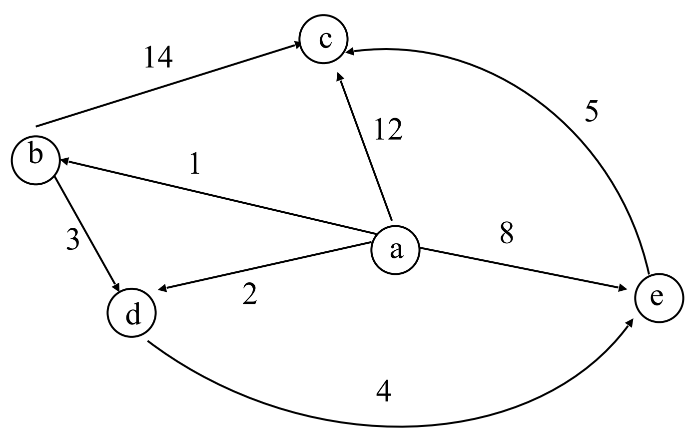
2ª Lista de Algoritmos e Grafos – 2015/1
Prof. Vinícius Gusmão

1) Diga se a afirmativa é verdadeira ou falsa.

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) A técnica da programação dinâmica por memoização permite resolver problemas numa abordagem *top-down*, de forma similar à formulação recursiva de muitos algoritmos, mas evitando que sub-problemas repetidos sejam resolvidos mais de uma vez.

2) Seja o digrafo *D* abaixo. Deseja-se determinar os caminhos de custo mínimo entre o vértice *a* (origem) 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. (PS.: A matriz de adjacências do grafo é dada, portanto obtê-la não é considerado como sendo a primeira iteração.)

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) 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*.