Estruturas de Dados e Algoritmos II
2ª Frequência e Exame

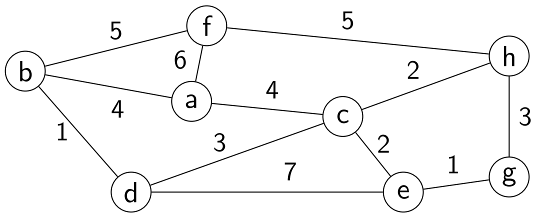
Departamento de Informática
Universidade de Évora

1 de Julho de 2021

Os símbolos à esquerda de cada pergunta identificam a prova ou provas a que ela pertence:
♣ assinala as perguntas do exame; ♢ assinala as perguntas da frequência.

Nas perguntas que pertencem ao exame **e** à frequência, o valor à esquerda de '/' corresponde à cotação da pergunta no exame, e o valor à direita de '/' corresponde à cotação da pergunta na frequência.

♣ ♢ **1.** [3/4 valores] Apresente uma ordem pela qual os arcos poderiam ser considerados durante uma aplicação do algoritmo de Kruskal ao grafo abaixo, indicando quais seriam incluídos na árvore de cobertura mínima e quais não seriam.



♢ **2.** [2 valores] Durante a aplicação do algoritmo de Floyd-Warshall, podem aparecer valores diferentes de 0 na diagonal principal de uma matriz com os pesos dos caminhos mais curtos? Qual a razão porque não podem ou, podendo, que outros valores lá podem aparecer e qual o seu significado?

♣ **3.** Na BigBang Desenvolvimento, está na altura de anunciar as promoções dos funcionários. Durante o resto do ano, o desempenho de cada um é observado e são estabelecidas prioridades de uns funcionários sobre outros, em relação à promoção de cada um. Depois, a empresa decide quantos serão os funcionários promovidos e escolhe-os, respeitando as prioridades estabelecidas. Estas prioridades são consistentes, no sentido em que nenhum funcionário tem, directa ou indirectamente, prioridade sobre si próprio.

O desempenho, ao longo do último ano, determinou que (anonimizando a informação) A tem prioridade sobre B e E, B tem prioridade sobre C e F, C tem prioridade sobre D, E tem prioridade sobre F, F tem prioridade sobre C, e G tem prioridade sobre E. Nestas condições, para E poder ser promovido, têm de ser pelo menos 3 os funcionários promovidos, visto que A e G têm prioridade sobre E. Por outro lado, D só será promovido se forem promovidos os 7 funcionários.

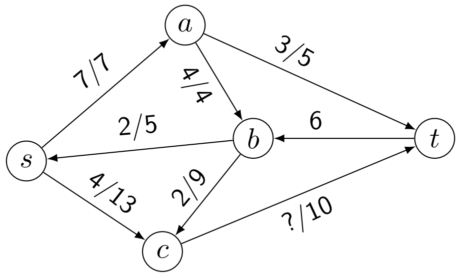
Pretende-se calcular, a partir das prioridades estabelecidas, quantos são os funcionários que têm de ser *obrigatoriamente* promovidos antes de um dado funcionário poder sê-lo também.

(a) [1,5 valores] Apresente o grafo que usaria para representar a informação relativa à situação descrita acima.

(b) [0,5 valores] Em termos de grafos, e tendo em conta aquele que apresentou na alínea anterior, a que corresponde o que se pretende contar?

(c) [2 valores] Descreva o algoritmo que utilizaria para obter o resultado pretendido. (Pode fazê-lo através de texto ou recorrendo a pseudo-código.)

4. Considere a rede de fluxos e o fluxo parcialmente representados:



♣ ◇ (a) [1 valor] Diga qual o valor de fluxo em falta (na ligação (c, t)).

♣ ◇ (b) [2/3 valores] Apresente a rede residual correspondente. (Se não respondeu à alínea anterior, pode usar o valor 5 como valor em falta.)

◇ **5.** [4 valores] Considere uma implementação de uma fila FIFO recorrendo a duas pilhas, S_1 e S_2: a operação ENQUEUE empilha o elemento em S_1; a operação DEQUEUE desempilha e devolve o elemento no topo de S_2, mas, se S_2 está vazia, primeiro desempilha cada elemento em S_1 e empilha-o em S_2 e só depois desempilha e devolve o elemento no topo de S_2.

Recorrendo ao método do potencial, calcule a complexidade amortizada da operação DEQUEUE. Use como função potencial a função:

$$\Phi(Q_i) = 2\, s_i,$$

onde s_i é o número de elementos na pilha S_1, no estado Q_i da fila, com $s_0 = 0$, e considere que o custo real de cada operação é número de elementos empilhados e desempilhados durante a execução da operação. (Assuma que há sempre, pelo menos, um elemento na fila, quando a operação DEQUEUE é invocada.)

♣ **6.** Considere a função recursiva $f_{Mn}[i]$, onde:

- n é um inteiro positivo;
- $M = (m_{11}\ m_{12}\ \ldots\ m_{1n}\ m_{21}\ \ldots\ m_{nn})$ é uma matriz de inteiros;
- $1 \leq i \leq n$.

$$f_{Mn}[i] = \begin{cases} m_{nn} & \text{se } i = n \\ \min\limits_{i<j\leq n} \left\{ f_{Mn}[j] + m_{ij} \right\} & \text{se } 0 < i < n \end{cases}$$

(a) [3,5 valores] Apresente o pseudo-código de um algoritmo iterativo que, dados um inteiro $n > 0$ e uma matriz não vazia de inteiros M, calcula e devolve o valor de $f_{Mn}[1]$.

(b) [0,5 valores] Estude (justificando) a complexidade temporal do seu algoritmo.

♣ **7.** [3 valores] Suponha que tem uma sequências de montes de moedas, que pode apanhar e ganhar. Todos os montes têm um número par de moedas e as regras para os apanhar são as seguintes:

i. Pode escolher, sem qualquer restrição, o primeiro monte a apanhar.

ii. Depois de apanhar um monte, só pode apanhar montes que estejam à direita desse.

iii. Quando apanha um monte com a_i moedas, não pode apanhar os $a_i/2$ montes que estão imediatamente à direita desse.

Seja (2 10 2 6 8) a sequência das moedas nos montes, da esquerda para a direita. Se apanhar, em primeiro lugar, o primeiro monte com 2 moedas, não pode apanhar o monte com 10. A seguir, pode apanhar o segundo monte com 2 moedas e, se o fizer, não poderá apanhar o monte com 6, mas poderá apanhar o com 8, e recolher um total de 12 moedas. Se optar por não apanhar o segundo monte com 2, só poderá apanhar o com 6 ou o com 8, perfazendo, no máximo, 10 moedas. Se começar pelo monte com 10, não poderá apanhar mais nenhum monte (não poderia apanhar os 5 montes seguintes, que são mais do que os 3 existentes).

Apresente *uma função recursiva* que, dados um inteiro positivo n e uma sequência não vazia de inteiros positivos pares $A = (a_1\ a_2\ \ldots\ a_n)$, representando a quantidade de moedas em cada um dos n montes, da esquerda para a direita, calcula o maior número de moedas que é possível recolher, apanhando os montes de acordo com as regras descritas acima.

Indique claramente o que representa cada uma das variáveis que utilizar e explicite a chamada inicial. (Note que não é pedido que escreva código.)

8. Seja $G = (V, E)$ um grafo não orientado e seja $p = v_1\, v_2 \ldots v_k$ um caminho simples em G, com $k < |V|$. Uma *extensão Hamiltoniana* de p é um caminho simples q em G, tal que $p\,q\,v_1$ é um circuito Hamiltoniano (um ciclo simples com comprimento $|V|$) em G. Por exemplo, no grafo da Pergunta 1, o caminho *chged* é uma extensão Hamiltoniana do caminho *bfa* (*bfachgedb* é um circuito Hamiltoniano no grafo).

O problema HAMILTON-EXT(G, p) é o problema de determinar se existe uma extensão Hamiltoniana do caminho simples p no grafo não orientado G.

◇ (a) [2 valores] Mostre que HAMILTON-EXT é um problema NP.

♣ ◇ (b) [3/4 valores] Seja HAMILTON(G) o problema de determinar se o grafo não orientado G tem um circuito Hamiltoniano.

Construa uma redução polinomial do problema HAMILTON para o problema HAMILTON-EXT.