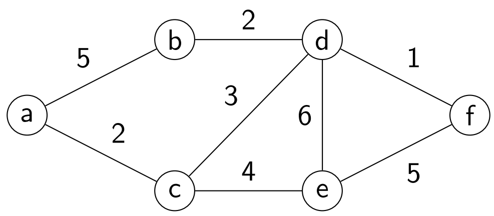
Estruturas de Dados e Algoritmos II
Exame de Recurso

Departamento de Informática
Universidade de Évora

20 de Julho de 2021

1. Considere o grafo da figura seguinte:



(a) [0,5 valores] Apresente um caminho *não* simples de comprimento 3.

(b) [2 valores] Apresente uma ordem pela qual os vértices do grafo poderiam ser introduzidos na fila, durante a execução de um percurso em largura sobre o grafo, a partir do vértice a.

(c) [2 valores] Considere que, durante uma aplicação do algoritmo de Prim ao grafo, os vértices que já estão na árvore em construção são a, c e e. Qual será o próximo vértice a ser incluído na árvore?

Apresente a árvore de cobertura parcial com os 4 vértices.

2. Numa viagem interplanetária quântica, quando o viajante sai de um planeta, pode ir parar a qualquer um de dois outros planetas fixos, sem ter controlo sobre a qual deles vai parar. Felizmente, no trajecto de um planeta para outro, o viajante tanto pode gastar tempo, como pode ganhar tempo. (Pode acontecer, por exemplo, o viajante passar por vários planetas, durante uma viagem, e chegar ao planeta destino 2 dias antes da data da partida.)

No sistema solar Sbeq, existem 5 planetas. Os destinos possíveis a partir de cada um deles são apresentados na tabela à direita, assim como o tempo gasto ou ganho em cada trajecto. Por exemplo, partindo do planeta A, tanto é possível chegar ao planeta B como ao D. A chegada a B dá-se 3 dias antes da partida, e a chegada a D dá-se 3 dias após a partida.

Planeta	Destino 1		Destino 2	
A	B	ganha 3 dias	D	gasta 3 dias
B	C	gasta 1 dia	E	ganha 5 dias
C	A	gasta 7 dias	B	gasta 1 dia
D	A	ganha 2 dias	C	gasta 2 dias
E	A	gasta 9 dias	C	gasta 4 dias

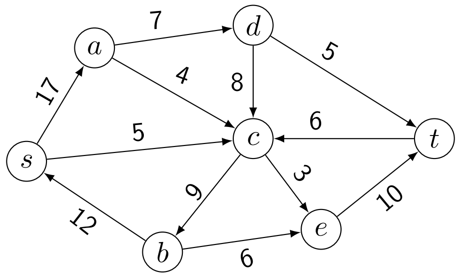
Sabendo que a partir de qualquer planeta é possível viajar para qualquer planeta, e dados os destinos possíveis a partir de cada um dos planetas, o tempo gasto ou ganho em cada trajecto, e um planeta, pretende-se determinar se é possível viajar desse planeta para ele próprio e chegar numa data anterior à da partida.

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

(b) [0,5 valores] Como classifica o grafo que apresentou na alínea anterior, em termos de conectividade?

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

3. [2,5 valores] Considere a rede de fluxos seguinte:



Efectue a primeira iteração do algoritmo de Edmonds-Karp sobre a rede apresentada. Apresente o caminho considerado, a sua capacidade residual, e a representação da rede com o fluxo calculado.

4. [2,5 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 o custo amortizado da operação ENQUEUE. 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 $s_0 = 0$. Considere que o custo real de cada operação é número de elementos empilhados e desempilhados durante a execução da operação.

5. Considere a função recursiva $f_A[i]$, onde:

- $A = (a_1\ a_2\ \ldots\ a_n)$, com $n > 0$, é uma sequência de inteiros positivos; e
- $1 \leq i \leq n$.

$$f_A[i] = \begin{cases} a_n & \text{se } i = n \\ \max\Big\{\ f_A[i+1],\ \ a_i + f_A[i + a_i + 1]\ \Big\} & \text{se } i + a_i < n \\ \max\Big\{\ f_A[i+1],\ \ a_i\ \Big\} & \text{se } i < n\ \wedge\ i + a_i \geq n \end{cases}$$

(a) [2,5 valores] Apresente o pseudo-código de uma função iterativa que, dada uma sequência não vazia de inteiros positivos $A = (a_1\ a_2\ \ldots\ a_n)$, com $n > 0$, calcula e devolve o valor de $f_A[1]$.

(b) [1 valor] Estude (justificando) as complexidades temporal e espacial do seu algoritmo.

6. [3 valores] Seja U um conjunto. Um conjunto S de subconjuntos de U *cobre* U se a reunião desses subconjuntos é U, e diz-se que S é uma *cobertura* de U. Por exemplo, se $U = \{1, 2, 3, 4\}$, então o conjunto de subconjuntos $S = \{\{1\}, \{1, 2, 3\}, \{2, 4\}\}$ é uma cobertura de U, de dimensão 3, e contém uma cobertura de U de dimensão 2.

Dados um conjunto U, um conjunto S de subconjuntos de U, que cobre U, e um inteiro positivo k, o *problema da cobertura de conjunto*, SET-COVER(U, S, k) é o problema de determinar se S contém uma cobertura de U, de dimensão não superior a k.

Seja $G = (V, E)$ um grafo não orientado. Uma *cobertura de vértices* de G é um subconjunto V' de V tal que qualquer que seja o arco $(u, v) \in E$, se tem que ou u ou v ou ambos estão em V'. Por exemplo, o conjunto $\{\mathsf{a}, \mathsf{b}, \mathsf{d}, \mathsf{e}\}$ é uma cobertura de vértices, de dimensão 4, do grafo da Pergunta 1.

Dados o grafo G e um inteiro positivo k', o *problema da cobertura de vértices*, VERTEX-COVER(G, k'), é o problema de determinar se existe uma cobertura de vértices de G, de dimensão não superior a k'.

Construa uma redução polinomial do problema VERTEX-COVER para o problema SET-COVER.

(Sugestão: Pense no conjunto que se quer cobrir.)