Mestrado Integrado em Engenharia Informática e Computação

Complementos de Programação e Algoritmos

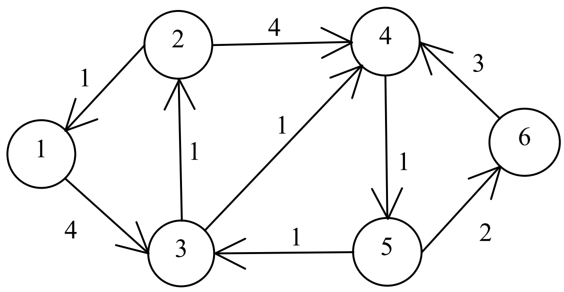
EXAME COM CONSULTA **11 Junho de 2008** DURAÇÃO: 2 horas

1. [7] Relativamente ao grafo dirigido da figura ao lado:

a) [1] indique se existe e, em caso de existir, apresente uma ordenação topológica dos vértices do grafo;

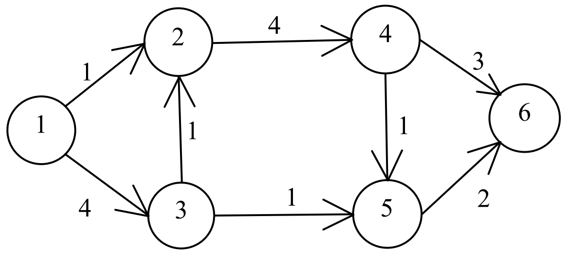
b) [1] indique um caminho mais curto (de peso total mínimo) do vértice 5 para o vértice 1;

c) [1] indique se o grafo é conexo ou fortemente conexo, justificando.



Relativamente à rede de transporte da figura ao lado

d) [1] indique o fluxo máximo que pode passar entre o nó 1 e o nó 6 (assumir que os pesos nas arestas indicam capacidades e apresentar o mesmo grafo com o fluxo que passa em cada aresta).

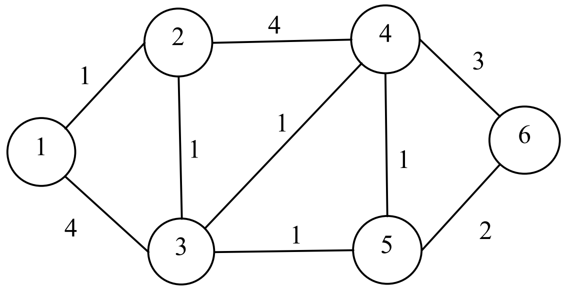


Relativamente ao grafo não dirigido da figura ao lado indique, justificando:

e) [1] uma árvore de expansão mínima;

f) [1] um caminho de Euler entre os vértices 2 e 5;

g) [1] todos os pontos de articulação existentes.



2. [7] Um condutor pretende efectuar uma viagem longa seguindo um percurso predefinido, ao longo do qual se encontram diversas bombas de abastecimento de combustível que praticam preços variados, e pretende planear os abastecimentos a efectuar de forma a minimizar o valor gasto em abastecimentos. Os abastecimentos devem ser planeados por forma a que a quantidade de combustível não desça abaixo de um nível mínimo de segurança.

Os dados de entrada para o problema são:

- m - capacidade **m**áxima do depósito, em litros;
- q - **q**uantidade de combustível existente inicialmente no depósito, em litros;
- s - quantidade mínima de **s**egurança de combustível no depósito, em litros;
- c - **c**onsumo de combustível em litros/km (suposto constante);
- $\{(d_1,p_1), \ldots, (d_n,p_n)\}$ - para cada bomba (i) existente ao longo do percurso, a distância ao ponto de partida em km (d_i) e o preço unitário praticado em euros/litro (p_i);
- d - **d**istância a percorrer.

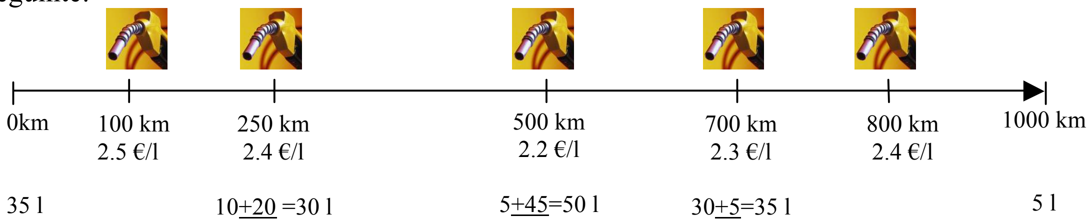
Os dados de saída pretendidas são:

- $q_1, \ldots, q_n$ - para cada bomba (i) existente ao longo do percurso, a quantidade (q_i) de combustível a abastecer em litros (0 no caso de não abastecer).

Por exemplo, dados os seguintes dados de entrada:

- m = 50 l q = 35 l s = 5 l c = 0,1 l/km d = 1000 km
- $\{(d_i,p_i)\}$ = {(100km, 2.5€/l), (250km, 2.4€/l), (500km, 2.2€/l),(700km, 2.3€/l), (800km, 2.4€/l)}

o plano de abastecimento óptimo (e quantidade de combustível) ao longo do percurso é esquematizado na figura seguinte:



a) [4] Conceba um algoritmo eficiente para obter o plano de abastecimento óptimo. Explique o algoritmo genericamente e apresente-o em pseudo código baseado em Java.

b) [1] Indique, justificando, que técnica(s) de concepção de algoritmos aplicou neste caso.

c) [1] Aplique o algoritmo passo a passo ao exemplo dado.

d) [1] Indique, justificando, a complexidade temporal do algoritmo. Em particular, indique que estruturas de dados e colecções do Java seriam usadas na implementação para conseguir a eficiência anunciada.

3. [6] Pretende-se determinar o caminho mais curto a seguir entre dois pontos numa rede viária, com a restrição de garantir que há pontos de abastecimento suficientes ao longo do percurso. Isto é, ao longo do percurso escolhido, a distância entre o ponto de partida e a primeira bomba de abastecimento, ente bombas consecutivas, e entre a última bomba e o ponto de chegada, não pode exceder a autonomia do veículo.

a) [4] Com base nos algoritmos em grafos estudados, conceba um algoritmo o mais eficiente possível para determinar o melhor caminho a seguir. É dado o mapa de estradas com as distâncias entre nós e a localização das bombas de abastecimento de combustível, bem como os pontos de partida e chegada no mapa. Apresentar os passos principais do algoritmo, explicando cada passo por palavras.

b) [2] Indique, justificando, a eficiência temporal e espacial do algoritmo.

Tópicos de resolução

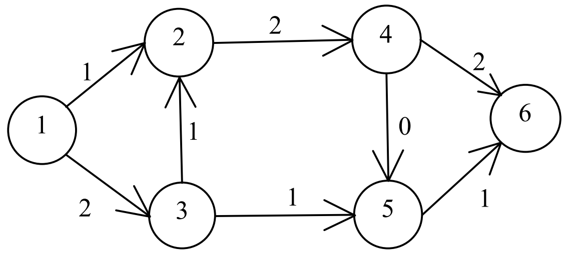
1.

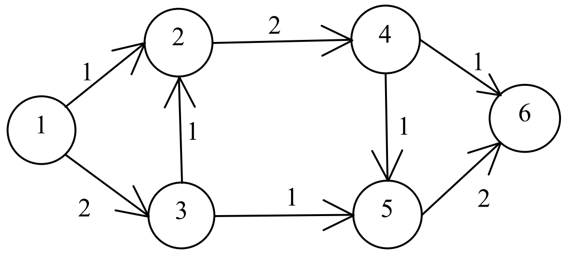
a) Não existe uma ordenação tipológica dos vértices, porque o grafo tem ciclos (1→3→2→1, etc.).

b) 5→3→2→1 (peso 3).

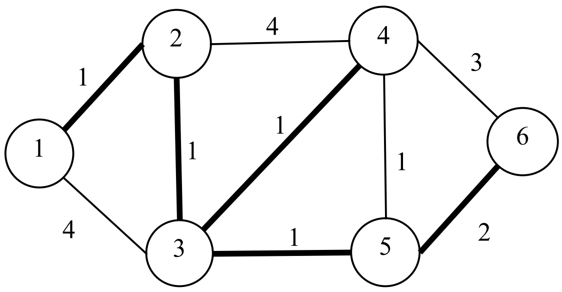
c) É fortemente conexo, pois existe um caminho dirigido entre quaisquer dois vértices.

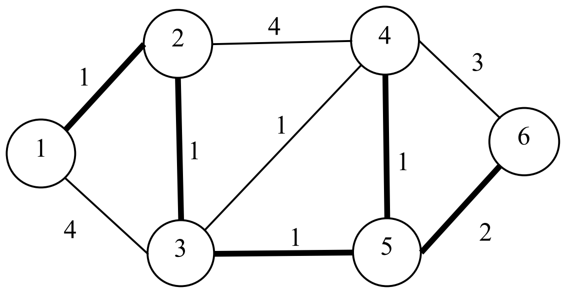
d) Fluxo máximo é 3. Grafo de fluxos pode ser um dos seguintes:

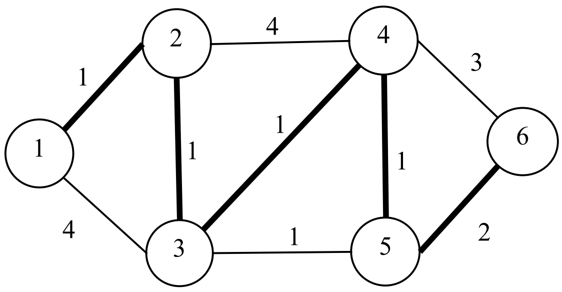




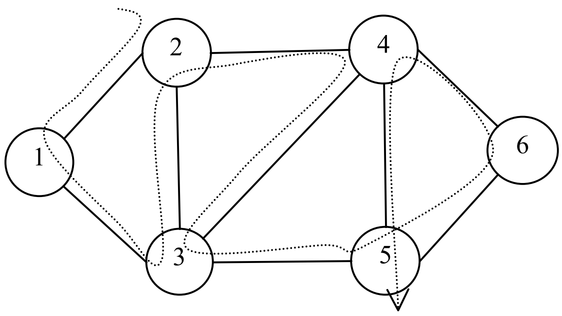
e) Soluções possíveis (indicar apenas uma):







f) Existe porque os vértices inicial e final têm grau ímpar e todos os outros têm grau par. Um caminho possível é: 2→1→3→2→4→3→5→6→4→5, conforme ilustrado na figura (existem mais possibilidades).



g) Não tem pontos de articulação.

2. a)

<u>Solução 1 - Algoritmo ganancioso</u>

1. Começar no início do percurso
   $d_{corrente} = 0;\quad q_{corrente} = q;\quad q_i = 0\ (i=1, ..., n);$
2. Se combustível for suficiente p/ chegar ao fim (acima nível segurança), não é preciso abastecer (termina)
   $d_{max} = d_{corrente} + (q_{corrente} - s) / c;$
   $\texttt{if } (d_{max} \geq d) \texttt{ return true;}$
3. Escolhe a bomba alcançável com preço mais baixo
   $i = \texttt{select } i \in \{1,...,n\} : p_i \texttt{ is min} \wedge d_i \in]d_{corrente}, d_{max}]$
4. Se não existir nenhuma bomba alcançável, falha
   $\texttt{if } (i == \texttt{null}) \texttt{ return false;}$
5. Avança até à bomba escolhida
   $q_{corrente} \mathrel{-}= (d_i - d_{corrente}) * c;\quad d_{corrente} = d_i;$
6. Se o destino for alcançável a partir desta bomba, abastece só o necessário para chegar ao destino
   $d_{max} = d_i + (m - s) / c;$
   $\texttt{if } (d_{max} \geq d)\ \{q_i = (d-d_i)*c + s - q_{corrente};\ q_{corrente} \mathrel{+}= q_i;\}$
7. Senão, se existe uma bomba mais barata alcançável a partir desta, abastece só o necessário para chegar lá
   $\texttt{else if } (\texttt{exists } j \in \{1,...,n\} : p_j < p_i \wedge d_j \in]d_i, d_{max}])$
   $\{q_i = (d_j - d_i)*c + s - q_{corrente};\ q_{corrente} \mathrel{+}= q_i;\}$
8. Senão, atesta
   $\texttt{else } \{q_i = m - q_{corrente};\ q_{corrente} = m;\}$
9. Continua no passo 2
   $\texttt{goto } 2$

<u>Solução 2 - Algoritmo de divisão e conquista</u>

1. Começar por considerar todo o percurso
   $d_{ini} = 0;\quad d_{fim} = d;\quad q_{ini} = q;\quad i_{min} = 1;\ i_{max} = n;\ q_i = 0\ (i=1,...,n);$
2. Se combustível for suficiente p/ chegar ao fim (acima nível segurança), não é preciso abastecer (termina)
   $d_{max} = d_{ini} + (q_{ini} - s) / c;$
   $\texttt{if } (d_{max} \geq d) \texttt{ return true;}$
3. Se não existir nenhuma bomba no percurso, falha
   $\texttt{if } (i_{min} > i_{max}) \texttt{ return false;}$
4. Escolhe a bomba com preço mais baixo no percurso
   $i = \texttt{select } i \in \{i_{min},...,i_{max}\} : p_i \texttt{ is min;}$
5. Abastece o máximo possível nessa bomba (chegando com mínimo e saindo com máximo)
   $q_{chegada} = \max(s, q_{ini} - (d_i - d_{ini}) * c);$
   $q_{saida} = \min(m, (d_{fim} - d_i) * c + s);$
   $q_i = q_{saida} - q_{chegada};$
6. Repete o procedimento (passos 2 a 7) do início até esta bomba
   $\texttt{repetir com } d_{fim} = d_i;\ i_{max} = i-1;$
7. Repete o procedimento (passos 2 a 7) desta bomba até ao fim
   $\texttt{repetir com } d_{ini} = d_i;\ i_{min} = i+1;\ q_{ini} = q_{saida};$

c) Mostrar o que se passa em cada iteração.

d) $O(n^2)$ - no máximo n (ou $2n$) iterações, cada uma $O(n)$ (procurar bomba mais barata)

3. a) Construir um grafo com as distâncias mais curtas entre todos os pares de bombas, bem como entre cada bomba e o início e fim (cada aresta representa um caminho no mapa original com uma distância). Eliminar arestas de distância superior à autonomia do veículo. Achar depois o caminho mais curto do ponto de partida ao ponto de chegada.