Departamento de Ciência de Computadores FCUP

(CC4010) Algoritmos 2022/2023

Folha Prática

1. Recordando que VERTEX COVER, na versão de decisão, é NP-completo, justifique que não é possível resolver problemas de programação inteira (genéricos) em **tempo polinomial** a menos que $P = NP$.

2. Supondo que P $\neq$ NP, classifique o problema enunciado em cada alínea como **membro ou não** das classes de problemas P, NP e NP-completos, **justificando**. Se for útil, na justificação pode usar conhecimentos sobre outros problemas.

a) Dados n e uma sequência $c_1, \ldots, c_n$ de inteiros positivos, e ainda um valor k, decidir se existe $I \subseteq \{1, \ldots, n\}$ tal que $|I| \geq k$ e $c_i \geq c_j$, qualquer que seja $i \in I$ e $j \in \{1, \ldots, n\}$, sendo $|I|$ o número de elementos de I.

b) Dado um grafo não dirigido $G = (V, E)$ e dada uma função $p : V \to \mathbb{Z}^+$ que associa um peso a cada vértice de G e ainda um valor $P \in \mathbb{Z}^+$, decidir se existe um conjunto de vértices $W \subseteq V$ tal que qualquer que seja a aresta $e \in E$, algum dos extremos de e pertence a W, e $\displaystyle\sum_{v \in W} p(v) \leq P$.

c) Dado um grafo não dirigido $G = (V, E)$, com $V = \{v_1, v_2, \ldots, v_n\}$, e dada uma permutação π de $\{1, 2, \ldots, n\}$, decidir se $(v_{\pi(1)}, v_{\pi(2)}, \ldots, v_{\pi(n)})$ é um ciclo de Hamilton em G.

d) Dada uma sequência $v_1, v_2, \ldots, v_n$, de n inteiros, decidir se está ordenada por ordem estritamente decrescente.

e) Dado um sistema de equações lineares $\mathtt{Ax} = \mathtt{b}$, com coeficientes inteiros e termos independentes inteiros, decidir se é satisfazível no conjunto dos inteiros não negativos.

f) Dadas duas sequências de inteiros $\mathtt{x}$ e $\mathtt{y}$, com $\mathtt{x} = (\mathtt{x_1}, \ldots, \mathtt{x_n})$ e $\mathtt{y} = (\mathtt{y_1}, \ldots, \mathtt{y_n})$, com possivelmente elementos repetidos, decidir se $\mathtt{x}$ é uma permutação de $\mathtt{y}$, isto é, se existe uma permutação π de $\{1, 2, \ldots, n\}$, tal que $x_k = y_{\pi(k)}$, para $1 \leq k \leq n$.

3. Seja **A** o problema de decidir se existe um caminho de s para t de comprimento maior ou igual a k num grafo $G = (V, E, \omega)$, dados G, s, t e k, com $\omega(e) \in \mathbb{Z}^+$, para $e \in E$. Seja **B** o problema de decidir se existe um caminho de s para t de comprimento menor ou igual a k em G, dados G, s, t e k.

Comente a veracidade ou falsidade da afirmação: *"Se* **P$\neq$NP***, o problema* **A** *não se pode reduzir polinomialmente ao problema* **B***, embora o problema* **B** *se possa reduzir polinomialmente ao problema* **A***"*.

4. Considere o problema do caixeiro viajante (TSP) num grafo não dirigido $G = (V, E, c)$, onde os custos dos ramos são **inteiros positivos quaisquer**. Recomendaria um algoritmo **polinomial** que resolvesse **qualquer** instância I do problema com garantia de que o custo $c(\gamma)$ da solução produzida para I fosse no máximo $2c(\gamma^\star)$, sendo $\gamma^\star$ uma solução ótima?

5. No problema BIN PACKING são dados n itens com pesos $p_1, \ldots, p_n$, tais que $0 < p_i \leq 1$, para todo i, e há que os distribuir por latas de capacidade unitária, usando o menor número de latas possível. Considere o algoritmo seguinte, sendo n e p dados do problema, e b e x *arrays* de n elementos e $n \geq 1$.

```
FIRSTFIT(p, b, n, x)
t = 1; b[1] = p[1]; x[1] = 1
for j = 2 to n do b[j] = 0
for i = 2 to n do
    j = 1
    while (j ≤ t and p[i] + b[j] > 1) do  j = j + 1
    if (j > t) then t = t + 1
    b[j] = b[j] + p[i];  x[i] = j
return t
```

a) Para a instância $n = 7$ e $p = [0.9, 0.4, 0.8, 0.3, 0.6, 0.5, 0.2]$, qual é o valor final de x, b e t? Conclua que FIRSTFIT não determina a solução ótima de BIN PACKING.

b) Justifique que FIRSTFIT determina em x uma distribuição admissível dos itens pelas latas (i.e., que não excede a capacidade das latas, podendo não ser ótima). Para tal, indique os invariantes de ciclo que nos permitem chegar a essa conclusão e verifique que são preservados em cada iteração do ciclo correspondente.

c) Determine a complexidade temporal assintótica de FIRSTFIT no melhor caso e no pior caso, para instâncias de n items. Caracterize instâncias de pior caso e de melhor caso.

d) Prove que FIRSTFIT produz uma solução aproximada, com fator de aproximação 2. Para isso, justifique que:

- Na solução obtida por FIRSTFIT, podemos ter no máximo uma lata que está com metade ou menos de metade da sua capacidade preenchida. Essa condição verifica-se ao longo da aplicação do algoritmo.
- Consequentemente, $\sum_{k=1}^{n} p_k > \frac{1}{2}(t-1)$, sendo t o número de latas na solução obtida por FIRSTFIT.
- Para o número de latas ótimo $t^\star$, tem-se $t^\star \geq \lceil \sum_{k=1}^{n} p_k \rceil$.

6. No problema PARTITION, são dados n inteiros positivos $a_1, a_2, \ldots, a_n$, e há que decidir se existe uma partição $\{S, T\}$ do conjunto de índices $\{1, 2, \ldots, n\}$ tal que $\sum_{i \in S} a_i = \sum_{i \in T} a_i$. Sabe-se que PARTITION é um problema **NP-completo**.

a) Justifique que as instâncias de PARTITION com $a_j > \sum_{j \neq i} a_i$, para algum j, são trivialmente decidíveis.

b) Dada uma instância de PARTITION, com $a_j \leq \sum_{j \neq i} a_i$, para todo j, definimos uma instância de BIN PACKING com $p_j = 2a_j / \sum_{i=1}^{n} a_i$, para todo j.

Justifique que tal instância de BIN PACKING se obtém em tempo polinomial e que requer pelo menos duas latas, sendo o ótimo igual a 2 se e só se a resposta para PARTITION for TRUE. Justifique que, usando essa **redução polinomial** se pode decidir PARTITION em tempo polinomial se algum dos algoritmos seguintes existir, e conclua que **não podem existir a menos que P=NP**:

(i) um algoritmo polinomial que calcule uma solução ótima para BIN PACKING;

(ii) um algoritmo de aproximação polinomial de razão c para BIN PACKING, com $c < 3/2$.

7. Considere uma variante de BIN PACKING, em que os pesos $p_1, \ldots, p_n$, são inteiros positivos e dispõe de m latas com capacidades possivelmente distintas, sendo c_k a capacidade da lata k, para $1 \leq k \leq m$

Justifique que o problema da minimização do número de latas é **NP-hard**.