Instituto Superior Técnico

Análise e Síntese de Algoritmos

Ano Lectivo 2019/2020 2º Teste

RESOLUÇÃO

I. (2,5 + 2,5 + 2,5 + 2,5 = 10,0 val.)

I.a) Considere o problema de compressão de dados de um ficheiro usando a codificação de Huffman. Indique o código livre de prefixo óptimo para cada carácter num ficheiro com 500 caracteres com a seguinte frequência de ocorrências: $f(a) = 18, f(b) = 13, f(c) = 12, f(d) = 5, f(e) = 7, f(f) = 45$. Quando constrói a árvore, considere o bit 0 para o nó com menor frequência.

Indique também o total de bits no ficheiro codificado.

	a	b	c	d	e	f
Codificação	111	110	101	1000	1001	0

Total Bits	1110

I.b) Considere a maior sub-sequência comum entre as duas strings $ACABACB$ e $AACABABC$ e calcule a respectiva matriz de programação dinâmica $c[i, j]$ para este problema, em que o índice i está associado à string $ACABACB$. Indique os seguintes valores: $c[4, 3]$, $c[4, 5]$, $c[7, 3]$, $c[5, 7]$, $c[7, 8]$.

c[4,3]	c[4,5]	c[7,3]	c[5,7]	c[7,8]
2	4	3	5	6

I.c) Considere o algoritmo de Knuth-Morris-Pratt com o seguinte texto $T = ababaabaaba$ e o padrão $P = bbabab$. Calcule a função de prefixo para o padrão P:

$P =$	b	b	a	b	a	b
i	1	2	3	4	5	6
$\pi[i]$	0	1	0	1	0	1

Indique ainda todas as posições da sequência de índices i que percorre o padrão P, ao aplicar o algoritmo sobre o texto T apenas quando o índice que percorre o texto é incrementado.

i	0	0	1	0	1	0	0	1	0	0	1	0

I.d) Considere o seguinte programa linear:

$$\begin{array}{rrrrcr}
\text{min} & -1x_1 & -\ x_2 & +\ x_3 & & \\
\text{s.a} & x_1 & +\ x_2 & +\ x_3 & \leq & 2 \\
 & -2x_1 & -2x_2 & -\ x_3 & \geq & -7 \\
 & x_1 & +2x_2 & +2x_3 & \leq & 4 \\
 & & & x_1, x_2, x_3 & \geq & 0
\end{array}$$

Indique o valor da função objectivo e o respectivo valor das variáveis básicas e não-básicas após uma única operação de pivot.

Z	x_1	x_2	x_3	x_4	x_5	x_6
-2	2	0	0	0	3	2

II. (2,5 + 2,5 + 2,5 + 2,5 = 10 val.)

II.a) O Eng. Caracol foi encarregado de apresentar uma proposta para a construção de postos de primeiros socorros ao longo da autoestrada AX, que começa no kilómetro 0 e termina no kilómetro k. O Eng. Caracol dispõe de uma lista de n locais candidatos. Cada local candidato, $1 \leq i \leq n$, é associado à sua distância ao kilómetro 0, d_i, e ao seu custo estimado de construção, c_i. Sabendo que dois postos de primeiros socorros consecutivos não podem estar a uma distância superior a D kilómetros, o objectivo do Eng. Caracol é determinar o conjunto de locais candidatos que satisfazem a restrição do problema pelo menor custo possível.

1. Seja $O(i)$ o custo da solução óptima para o troço da autoestrada AX entre o kilómetro 0 e o local candidato i, que atribui necessariamente um posto de primeiros socorros ao local candidato i. Defina $O(i)$ recursivamente, completando os campos abaixo.
$$O(i) = \begin{cases} \boxed{} & \text{se } i > 1 \\ \boxed{} & \text{caso contrário} \end{cases}$$
2. Complete o template de código em baixo que calcula a quantidade $O(i)$ para $1 \leq i \leq n$ e indique a respectiva complexidade assimptótica.

```
FindO(c[1..n], d[1..n])
  let O[1..n] be a new vector of size n
  O[1] = c[1]
  for i = 2 to n do
    [                    ]
  endfor
  return O
```

3. Explique como determinar o custo da melhor solução a partir do vector O.

Solução:

1. $$O(i) = \begin{cases} \min\{O(j) + c_i \mid j < i \ \wedge \ d_i - d_j \leq D\} & \text{se } i > 1 \\ c_1 & \text{caso contrário} \end{cases}$$

2. Em baixo o código completo:

```
Find0(c[1..n], d[1..n])
  let O[1..n] be a new vector of size n
  O[1] = c[1]
  for i = 2 to n do
    O[i] = ∞
    for j = i − 1 to 1 do
      if (d[i] − d[j] < D) then
        if (O[i] > O[j] + c[i]) then
          O[i] = O[j] + c[i]
        endif
      endif
    endfor
  endfor
  return C
```

   Complexidade: $O(n^2)$

3. A partir do vector O, calculamos um vector O' da seguinte maneira:

$$O'(i) = \min\{O(j) \mid j \leq i \ \wedge \ d_i - d_j \leq D\}$$

   O valor mínimo pretendido corresponde a $O'(n)$, que pode ser calculado em tempo linear.

II.b) Considere a seguinte variação do algoritmo de Rabin-Karp para emparelhamento de cadeias de caracteres sobre o alfabeto $\Sigma = \{a, b, c\}$, que usa a função de hash:

$$h(x_1...x_n) = (\, \alpha(x_1) + ... + \alpha(x_n)\,)\, \mathbf{mod}\ 5$$

onde: $\alpha(a) = 1$, $\alpha(b) = 2$ e $\alpha(c) = 3$. Seja $T = abab^2ab^3 \ldots ab^{n-1}ab^n$ a string de texto a processar. Calcule o número de *hits* espúrios, em função de n, gerados ao procurar os seguintes padrões em T (deve apresentar os cálculos): Nota: $\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$

1. $P_1 = abb$
2. $P_2 = abc$

Solução:

1. Observamos que $h(abb) = 0$. Os outros padrões com hash 0 que ocorrem em T são bab e bba. Obtemos a expressão:

$$\left(\sum_{i=2}^{n-1} 2 \right) + 1 = 2n - 3$$

2. Observamos que $h(abc) = 1$. O outro padrão com hash 1 que ocorre em T é bbb. Obtemos a expressão:

$$\begin{aligned} (\textstyle\sum_{i=3}^{n} i - 2) &= (\textstyle\sum_{i=1}^{n} i) - 3 - 2(n-2) \\ &= (\textstyle\sum_{i=1}^{n} i) - 2n + 1 \\ &= \frac{n^2-3n+2}{2} \end{aligned}$$

II.c)

Considere o seguinte programa linear:

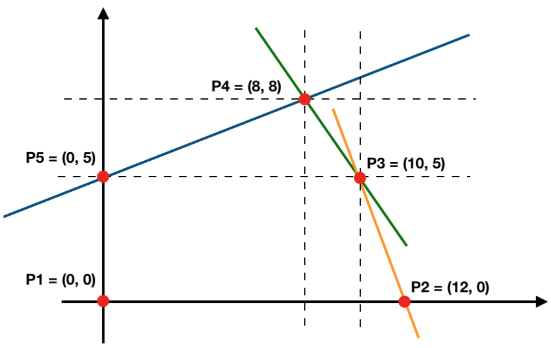
$$\begin{array}{rrrcl} \max & 2x_1 & +x_2 & & \\ \text{s.a} & 5x_1 & +2x_2 & \leq & 60 \\ & 3x_1 & +2x_2 & \leq & 40 \\ & -3x_1 & +8x_2 & \leq & 40 \\ & & x_1, x_2 & \geq & 0 \end{array}$$

1. Desenhe o conjunto exequível e resolva geometricamente o programa linear (indique tanto o valor máximo como as coordenadas onde esse valor é atingido).

2. Formule o programa linear dual e calcule a respectiva solução a partir da solução do programa primal (indique tanto o valor mínimo como as coordenadas onde esse valor é atingido). Nota: pode obter o valor das coordenadas resolvendo o sistema de equações do problema dual colocando a 0 as variáveis que correspondem a restrições não activas.

Solução:

1. Representamos a região exequível no diagrama em baixo.



   O Teorema Fundamental da Programação Linear estabelece que o valor óptimo da função objectivo, a existir, ocorre num vértice da região exequível. Assim sendo, concluímos que o valor óptimo é 25 e ocorre no ponto $P_3 = (10, 5)$.

2. O programa linear dual é definido em baixo:

$$\begin{array}{rrrrcl} \min & 60y_1 & +40y_2 & +40y_3 & & \\ \text{s.a} & 5y_1 & +3y_2 & -3y_3 & \geq & 2 \\ & 2y_1 & +2y_2 & +8y_3 & \geq & 1 \\ & & y_1, y_2, y_3 & & \geq & 0 \end{array}$$

   Do Teorema da Dualidade Forte concluímos que o valor mínimo do programa dual coincide com o valor máximo do programa primal, 25. Da inspecção da geometria do programa primal, concluímos que as restrições activas no vértice da solução correspondem às variáveis y_1 e y_2 do problema dual. Segue, por isso, que $y_3 = 0$ no ponto óptimo do problema dual. Resolvendo o sistema:

$$\begin{cases} 5y_1 + 3y_2 = 2 \\ 2y_1 + 2y_2 = 1 \end{cases}$$

   concluímos que o valor mínimo do programa dual se encontra no ponto $(1/4, 1/4, 0)$.

II.d) Uma matriz de incompatibilidades é uma matriz quadrada cujas células guardam valores decimais entre 0 e 1. Intuitivamente, dada uma matriz de incompatibilidades M, $n \times n$, a célula M_{ij} guarda a incompatibilidade entre os índices i e j; $M_{ij} = 0$ se i e j são completamente compatíveis e $M_{ij} = 1$ se i e j são completamente incompatíveis. Dado um sub-conjunto de índices $I \subseteq \{1, ..., n\}$, o nível de incompatibilidade do conjunto é dado por: $\sum_{i,j \in I} M_{ij}$. O problema das incompatibilidades define-se formalmente da seguinte maneira:

$$\textbf{Incompat} = \{\langle M, k, v\rangle \mid M \text{ contém um sub-conjunto de índices de tamanho } k \text{ e incompatibilidade igual ou inferior a } v\}$$

1. Mostre que o problema **Incompat** está em **NP**.
2. Mostre que o problema **Incompat** é NP-difícil por redução a partir do problema **ISet**, que é sabido tratar-se de um problema NP-completo e que se define em baixo. Não é necessário provar formalmente a equivalência entre os dois problemas; é suficiente indicar a redução e a respectiva complexidade.
   Pista: Dado um grafo G indique como construir uma matriz de incompatibilidades cujos índices correspondem aos vértices de G tendo em conta o problema **ISet**.

Problema ISet: Seja $G = (V, E)$ um grafo não dirigido; dizemos que $V' \subseteq V$ é um conjunto de vértices independentes em G se e apenas se $\forall u, v \in V'.(u, v) \notin E$. O problema **ISet** define-se formalmente da seguinte maneira:

$$\textbf{ISet} = \{\langle G, k\rangle \mid G \text{ contém um conjunto de vértices independentes de tamanho } k\}$$

Solução:

1. O algoritmo de verificação recebe como input uma possível instância $\langle M, k, v\rangle$ e um conjunto de índices I (o certificado). O algoritmo tem de verificar que $|I| = k$ e que $\sum_{i,j \in I} M_{ij} \leq v$. Observamos os certificados têm tamanho $O(n)$ e que a verificação se faz em tempo $O(n^2)$, o tempo de calcular o somatório.

2. Dada uma possível instância $\langle G, k\rangle$ do problema **ISet**, começamos por construir uma matriz de incompatibilidades M_G cujos índices correspondem aos vértices de G. Para tal, admitimos que $|V| = n$ e que os vértices de V estão numerados de 1 a n, sendo v_i o i-ésimo vértice. Assim sendo, definimos a matriz M_G como se segue:
$$(M_G)_{ij} = \begin{cases} 1 & \text{se } (v_i, v_j) \in E \\ 0 & \text{caso contrário} \end{cases}$$
   Uma vez estabelecida a matriz M_G, a redução é definida da seguinte maneira:
$$f(\langle G, k\rangle) = \langle M_G, k, 0\rangle$$
   - Equivalência a estabelecer: $\langle G, k\rangle \in \textbf{ISet} \iff \langle M_G, k, 0\rangle \in \textbf{Incompat}$
   - Complexidade da redução: $O(|V|^2)$.