Aula Prática 6

ASA 2024/2025

Q1 (CLRS Ex. 16.2-4) Professor Gekko has always dreamed of inline skating across North Dakota. He plans to cross the state on highway U.S. 2, which runs from Grand Forks, on the eastern border with Minnesota, to Williston, near the western border with Montana. The professor can carry two liters of water, and he can skate m miles before running out of water. (Because North Dakota is relatively flat, the professor does not have to worry about drinking water at a greater rate on uphill sections than on flat or downhill sections.) The professor will start in Grand Forks with two full liters of water. His official North Dakota state map shows all the places along U.S. 2 at which he can refill his water and the distances between these locations. The professor's goal is to minimize the number of water stops along his route across the state. Give an efficient method by which he can determine which water stops he should make. Prove that your strategy yields an optimal solution, and give its running time.

Solução:

Para minimizar o número de paragens para abastecimento de água, maximizamos a distância que percorremos com a água existente.

Sub-estrutura óptima:

- assumimos que as localizações estão dispostas ao longo de uma recta. Em particular, podemos assumir que é conhecida a distância do ponto de partida a todas as n localizações, $D = \{d_0, d_1, d_2, \ldots, d_{n-1}, d_n\}$.
- solução greedy tenta avançar o máximo possível com a água existente e abastece nesse local. Depois resolve o sub-problema a partir desse local.

Highway-Greedy(D, C$_m$)

```
last ← 0
S ← {0}
for i ← 1 to |D| - 1 do
  if d_{i+1} − last > C_m then
    S ← S ∪ {i}
    last ← d_i
  end if
end for
return S
```

Q2 (CLRS Ex. 16.2-5) Describe an efficient algorithm that, given a set $\{x_1, x_2, \ldots, x_n\}$ of points on the real line, determines the smallest set of unit-length closed intervals that contains all of the given points. Argue that your algorithm is correct.

Solução:

$X = \{x_1, \ldots, x_n\}$ é o conjunto de n pontos na recta real.

Queremos o menor número de intervalos de tamanho 1 não contíguos, que contenham todos os n pontos.

- assumir pontos ordenados, caso contrário ordenar previamente
- a solução greedy é óptima dado que a cada passo consideramos o maior número de pontos do lado esquerdo da recta, e o ponto x_1 tem de pertencer a algum intervalo

IntervalSet(X)

```
limit ← x1 + 1
S ← {[x1, limit]}
for i ← 2 to |X| do
  if xi > limit then
    limit ← xi + 1
    S ← S ∪ {[xi, limit]}
  end if
end for
return S
```

Q3 (T2 08/09 II.3) Considere o seguinte conjunto de caracteres e as respectivas frequências:

a:49   b:23   c:12   d:15   e:7   f:4   g:10

Atendendo às frequências indicadas, utilize o algoritmo de Huffman para determinar a codificação de Huffman, indicando para cada caracter a palavra binária resultante. Quando constrói a árvore, considere o bit a 0 no ramo para o nó com menor frequência.

Solução:

a	b	c	d	e	f	g
0	111	100	101	11011	11010	1100

Q4 (R2 08/09 II.3) Considere que pretende planear a aterragem de um conjunto de vôos numa das pistas de um aeroporto, durante uma hora de um determinado dia. Um vôo i, ocupa a pista desde o minuto previsto para a sua aterragem, a_i, até que o procedimento de aterragem esteja completo. A duração (em minutos) do procedimento de aterragem de cada vôo i é dada por d_i. Assim, no instante $a_i + d_i$, imediatamente após a aterragem do vôo i, a pista encontra-se novamente livre. Considerando a informação dos vôos fornecida na tabela abaixo, assinale com $\times$ os vôos aos quais deve ser dada indicação de aterragem na referida pista, por forma a maximizar o número total de aterragens que nela serão efectuadas, durante o período considerado. <u>Nota:</u> Se, por exemplo, um vôo aterrar ao minuto 15 e o procedimento de aterragem demorar 5 minutos, ao minuto 20 poderá aterrar outro vôo.

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
a_i	00	06	10	13	17	20	23	25	28	31	33	36	39	40	43	50	53
d_i	5	5	3	6	1	3	2	6	2	4	5	3	5	2	4	6	2

Solução:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
x	x			x	x	x		x	x		x		x	x		x

Q5 (T2 14/15 II.b) Considere que tem duas caixas e n objectos onde cada objecto i ($1 \leq i \leq n$) tem peso p_i e valor v_i. Considere ainda que cada caixa tem capacidade para suportar até K unidades de peso, sendo que o somatório do peso dos n objectos é superior a $2K$. Ou seja, $\sum_{i=1}^{n} p_i > 2K$.

Supondo que consegue seleccionar qualquer fracção dos n objectos, elabore um algoritmo que permita seleccionar os objectos a colocar em cada uma das duas caixas tal que o valor colocado em cada uma das caixas seja maximizado e o peso dos objectos seja o mesmo em ambas as caixas.

Indique a complexidade do algoritmo proposto.

Solução: Este problema é equivalente ao problema da mochila fraccionária, pelo que tem uma solução *greedy*.

Começamos por fazer a ordenação dos n objectos de forma não crescente do valor por unidade de peso. Seguindo essa ordem, vamos colocando sucessivamente 50% de cada objecto em ambas as caixas. Quando atingirmos um objecto i tal que $p_i/2$ excede a capacidade residual das caixas, seleccionamos apenas a percentagem suficiente para encher cada uma das duas caixas e o algoritmo termina.

Este algoritmo tem complexidade $O(n\ lg\ n)$ devido ao procedimento de ordenação dos n objectos. A selecção dos objectos a colocar nas caixas é $O(n)$.

Q6 (R2 14/15 II.b) Considere o problema de compressão de dados de um ficheiro usando a codificação de Huffman. Indique o código livre de prefixo óptimo para cada caractere num ficheiro com 10000 caracteres com as seguintes frequências de ocorrência: $f(a) = 1, f(b) = 15, f(c) = 8, f(d) = 20, f(e) = 2, f(f) = 24, f(g) = 30$. Quando constrói a árvore, considere o bit 1 para o nó com menor frequência.

Indique o total de bits no ficheiro codificado.

Solução:

a	b	c	d	e	f	g	Total Bits
01111	010	0110	11	01110	10	00	24000

Q7 (R2 15/16 II.a) Considere que é caixa num hipermercado. Quando os clientes pagam em numerário, tem que usar notas e moedas para perfazer o troco a dar ao cliente.

Suponha que tem um conjunto de notas e moedas denominadas $1 \dots n$ cujos valores são $d_1 \dots d_n$ tal que $d_1 = 1$ e $d_i \geq 2 * d_{i-1}$ onde $2 \leq i \leq n$.

Considere que tem acesso a um número infinito de cada uma das n denominações. Indique um algoritmo greedy que use as n denominações para perfazer um troco de K por forma a que o número de notas e moedas usadas no troco seja mínimo.

Se a condição $d_i \geq 2 * d_{i-1}$ onde $2 \leq i \leq n$ não se verificar, o seu algoritmo ainda produz o número mínimo de moedas? Justifique ou indique um contraexemplo.

Solução: Supondo que desejamos perfazer um troco de K, iteramos sobre as diferentes denominações começando na denominação n até à primeira. Para cada denominação i, escolhemos usar m_i moedas onde $m_i = K\ div\ d_i$ (onde div denota o operador para a divisão inteira) e actualizamos K para $K - m_i * d_i$. O número total de moedas seria $\sum_{i=1}^{n} m_i$

O algoritmo greedy pode não produzir a solução óptima se a condição não se verificar.

Se $n = 3$ e $d_1 = 1, d_2 = 4, d_3 = 6$, o algoritmo greedy indica 3 moedas para perfazer um troco de 8 quando a solução óptima seria apenas 2 moedas.

Q8 (R2 16/17 I.a) Considere o problema de compressão de dados de um ficheiro usando a codificação de Huffman para um ficheiro com as seguintes as ocorrências indicadas na seguinte tabela.

	a	b	c	d	e
f	21	20	19	18	22

Indique para cada caracter a palavra binária resultante. Quando constroi a árvore considere o bit 0 associado ao ramo com menor frequência.

Solução:

a	b	c	d	e
01	00	111	110	10