

UNIVERSIDADE DE COIMBRA FACULDADE DE CIÊNCIAS E TECNOLOGIA

Departamento de Engenharia Informática

Trabalho nº 1 de Algoritmos e Estruturas de Dados

2013-2014 - 2° Semestre

Data-limite de submissão no Mooshak:

- Tarefa A: 23 de Fevereiro de 2014, 23h59.
- Tarefa B: 23 de Fevereiro de 2014, 23h59.
- Tarefa C: 02 de Marco de 2014, 23h59.

Data-limite de entrega do relatório:

05 de Março de 2014, 14h00.
Em papel, manuscrito e entregue no cacifo do respectivo docente das aulas TP.

<u>Nota Importante</u>: A fraude denota uma grave falta de ética e constitui um comportamento não admissível num estudante do ensino superior e futuro profissional licenciado. Qualquer tentativa de fraude pode levar a anulação da componente prática tanto do facilitador como do prevaricador.

Objectivos: Análise de Complexidade.

Notas Importantes:

- O relatório do trabalho deve ser realizado, MANUSCRITO, no formulário que será oportunamente fornecido.
- A não entrega do relatório do trabalho no prazo-limite estabelecido implica que o aluno tem uma classificação de ZERO na avaliação deste trabalho.

O dilema do diretor de publicidade - que anúncios escolher?

- Tarefa A Solução baseada em abordagem sôfrega
- Tarefa B Solução baseada em análise de combinações
- Tarefa C Solução baseada em programação dinâmica

Descrição do Problema

O diretor de publicidade da TVI pretende maximizar os valores de venda de publicidade.

O tempo máximo de publicidade disponível em cada período é conhecido à partida e determinado por imposições legais e de programação, que não podem ser alteradas.

Cada anunciante propõe-se pagar um determinado valor 'V' pela passagem do seu anúncio, que tem uma duração de 'T' segundos.

Pretende-se maximizar o retorno para a TVI, escolhendo a combinação de anúncios, para o tempo máximo disponível, que maximize o retorno financeiro. No entanto, como nem sempre os anúncios mais longos são os que têm um preço mais elevado, a escolha não é fácil.

Para o ajudar, construa uma aplicação que:

- recebe como *input* o tempo disponível para anúncios e uma lista com o valor das *N* propostas de anúncios recebidas (incluindo, para cada uma, o valor 'V' que o anunciante se propõe pagar e a duração 'T' do anúncio).
- apresenta como *output* o maior valor possível de retorno financeiro, para a TVI.

Formato do *Input*:

A primeira linha do *input* indica:

- o tempo disponível (em segundos)
- e o número total de anúncios a considerar (N).

As N linhas seguintes indicam:

• o valor 'V' que o anunciante se propõe pagar e a duração do anúncio ('T').

Por motivos de simplificação, é garantido que os anúncios estão sempre ordenados por ordem decrescente do seu valor.

Exemplo A: para um tempo total disponível de 18 segundos, 4 ofertas de anúncios com os valores {{30, 5} {20, 12}{6, 6}{5 10}} {Valor, Tempo}

Input:

18 4		
30 5		
20 12		
6 6		
5 10		

Observações:

18 segundos no total, 4 ofertas anúncio de 30 Euros, dura 5 segundos anúncio de 20 Euros, dura 12 segundos anúncio de 6 Euros, dura 6 segundos anúncio de 5 Euros, dura 10 segundos

Exemplo B: um tempo total disponível de 5 segundos, com 4 ofertas de anúncios, com os valores $\{\{6, 5\}, \{5, 4\}, \{4, 3\}, \{3, 2\}\}$

5 4 6 5			
6 5			
5 4			
4 3			
3 2			

5 segundos no total, 4 ofertas anúncio de 6 Euros, dura 5 segundos anúncio de 5 Euros, dura 4 segundos anúncio de 4 Euros, dura 3 segundos anúncio de 3 Euros, dura 2 segundos

Exemplo C: um tempo total disponível de 7 segundos, com 3 ofertas de anúncios com os valores $\{\{10, 10\} \{10, 9\} \{8, 30\}\}\$

7 3			
10 10			
10 9			
8 30			

Tarefa A :: Solução baseada numa abordagem sôfrega

Na Tarefa A deverá usar uma abordagem sôfrega baseada na seguinte heurística:

- As propostas de anúncios são dispostas por ordem decrescente de valor.
- O anúncio mais caro é selecionado (se a sua duração for inferior ou igual ao tempo disponível).
- De seguida a operação é sucessivamente repetida (selecionar o próximo anúncio mais caro, caso a sua duração seja inferior ou igual ao tempo remanescente disponível).
- O *output* corresponde à diferença entre a soma do valor de cada subconjunto.

Esta solução é simples e aparentemente eficiente, ainda que nem sempre produza a melhor solução possível – como poderá constatar no output correspondente ao exemplo B, para o qual até havia uma outra solução mais vantajosa.

Output correspondente ao Exemplo A (anúncios #1 e #2)

50

Output correspondente ao Exemplo B (anúncio #1)

6

Output correspondente ao Exemplo C (nenhum anúncio pode ser incluído)

0

Tarefa B :: Solução baseada numa abordagem de "força-bruta"

Na Tarefa B deverá usar uma abordagem baseada em "força-bruta", que estime todas as formas possíveis de repartir os objetos, de modo a determinar qual a menor diferença que pode ser encontrada na repartição do tesouro.

Esta abordagem poderá usar recursividade para explorar as várias formas possíveis de repartir os objetos, tendo em conta que:

- O 1º anúncio pode ou não fazer parte da melhor solução.
- Para cada uma das duas opções relativamente ao 1º anúncio (fazer ou não parte da solução), o 2º anúncio pode igualmente fazer ou não parte da melhor solução, dando um total de 4 alternativas possíveis para distribuir os primeiros 2 anúncios.

• E assim sucessivamente... até chegar ao último objeto.

Dado que esta abordagem é exaustiva, existe a garantia de encontrar efetivamente a menor diferença possível entre os 2 conjuntos, como pode ser verificado no exemplo B.

Output correspondente ao Exemplo A (anúncios #1 e #2)

50

Output correspondente ao Exemplo B (anúncios #3 e #4)

7

Output correspondente ao Exemplo C (nenhum anúncio pode ser incluído)

0

Caso pretenda, poderá introduzir algumas optimizações nesta abordagem (por exemplo parar quando a solução que está a explorar já ultrapassa o tempo máximo disponível), que poderá OPCIONALMENTE introduzir na sua solução e mencionar/analisar no relatório final.

Tarefa C :: Solução baseada em programação dinâmica

Na Tarefa C deverá usar uma abordagem baseada em programação dinâmica, tirando partido das seguintes observações:

- Uma solução sem qualquer anúncio tem valor "zero".
- Se o tempo máximo admissível for 5 segundos, por exemplo, a melhor solução terá um tempo total ocupado com os anúncios de: 0, 1, 2, 3, 4 ou 5 segundos.
- A abordagem exaustiva da Tarefa B repete várias vezes cálculos que poderiam ser dispensados, caso pudéssemos seguir uma abordagem "bottom-up", estimando primeiro as soluções para conjuntos de anúncios mais pequenos e armazenando os valores obtidos para depois reutilizar na procura das soluções com conjuntos maiores.

Para operacionalizar este raciocínio, pode usar uma *matriz* auxiliar V, de tipo *inteiro*, com "Tempo máximo + 1" colunas e "total de anúncios + 1" linhas. O conteúdo desta matriz será o máximo valor obtido para cada subconjunto e para cada limite de tempo

Esta matriz vai ser usada para armazenar os cálculos das melhores soluções com conjuntos de anúncios sucessivamente maiores (primeiro sem nenhum anúncio, depois apenas com o 1º anúncio, depois com os primeiros 2 anúncios e assim sucessivamente).

Recordando o exemplo B, vamos então construir a seguinte matriz:

Anúncio / Tempo total		0	1	2	3	4	5
Sem anúncios	0						
1 anúncio { <u>6 5</u> }	1						
2 anúncios {6 5}{ <u>5 4</u> }	2						
3 anúncios { <u>4 3</u> }	3						
4 anúncios {3 2}	4						

De seguida podemos preencher desde já todas as soluções sem anúncios (1ª linha da matriz) e todas as soluções com tempo "0" (1ª coluna da matriz).

Anúncio / Tempo total		0	1	2	3	4	5
Sem anúncios	0	0	0	0	0	0	0
1 anúncio {6 5}	1	0					
2 anúncios {5 4}	2	0					
3 anúncios {4 3}	3	0					
4 anúncios {3 2}	4	0					

De seguida podemos preencher todas as soluções com o 1º anúncio (6€ 5s), usando o seguinte mecanismo:

- Para todas as colunas com um tempo total inferior ao tempo do anúncio o valor da linha anterior é mantido (o anúncio não pode ser incluído na solução final para esse "tempo").
- Para as restantes colunas existem 2 opções:
 - Caso a soma do valor que esteja na coluna (i "tempo do anúncio") com o valor do anúncio seja superior ao valor da coluna i na linha anterior a coluna i toma como valor essa soma (ou seja, o anúncio faz parte da melhor solução

intermédia em análise para i segundos).

 Caso contrário o valor da coluna i será igual ao valor da coluna i na linha anterior (o anúncio ainda caberia no tempo disponível mas não faz parte da melhor solução intermédia em análise para i segundos).

Anúncio / Tempo total		0	1	2	3	4	5
Sem anúncios	0	0 -	0	0	0	0	0
1 anúncio {6 5}	1	0	0	0	0	0	6€
2 anúncios {6 5}{5 4}	2	0					
3 anúncios {4 3}	3	0					
4 anúncios {3 2}	4	0					

Esta operação é depois sucessivamente repetida para os restantes anúncios:

Anúncio / Tempo total		0	1	2	3	4	5
Sem anúncios	0	0	0	0	0	0	0
1 anúncio {6 5}	1	0	0	0	0	0	6€
2 anúncios {5 4}	2	0	0	0	0	5€	6€
3 anúncios {4 3}	3	0					
4 anúncios {3 2}	4	0					

Anúncio / Tempo total		0	1	2	3	4	5
Sem anúncios	0	0	0	0	0	0	0
1 anúncio {6 5}	1	0	0	0	0	0	6€
2 anúncios {5 4}	2	0 —	0	0	0	5 €	6€
3 anúncios {4 3}	3	0	0	0	4 €	5€	6€
4 anúncios {3 2}	4	0					

Anúncio / Tempo total		0	1	2	3	4	5
Sem anúncios	0	0	0	0	0	0	0
1 anúncio {6 5}	1	0	0	0	0	0	6€
2 anúncios {5 4}	2	0	0	0	0	5€	6€
3 anúncios {4 3}	3	0	0	0	4€	5€	6€
4 anúncios {3 2}	4	0	0	3 €	4 €	5 €	7€ ₩

O resultado final que maximiza as receitas para a TVI corresponde ao valor máximo registado na última linha. No caso do exemplo é a combinação entre os anúncios #3 e #4, que dura 5 segundos e tem um valor de 7 Euros, mas noutros casos poderá corresponder a combinações com menor duração (isto é, a outras colunas dessa última linha).

Tendo em conta os três exemplos apresentados, os *outputs* esperados são os seguintes:

Output correspondente ao Exemplo A (anúncios #1 e #2)

50

Output correspondente ao Exemplo B (anúncios #3 e #4)

7

Output correspondente ao Exemplo C (nenhum anúncio pode ser incluído)

0