

DEI/FCTUC – Universidade de Coimbra

Algoritmos e Estruturas de Dados, 2010/2011

Enunciado do Trabalho Prático TP2

Técnicas de Projeto de Algoritmos:

- Seleção Sôfrega
- Recursão
- Programação Dinâmica

Exercícios

- Problema A – Solução Sôfrega
- Problema B – Solução Recursiva
- Problema C – Solução com Programação Dinâmica

Prazo limite de submissão no *Mooshak*: 22/Fevereiro/2011, 12h00 (Problema A)
01/Março/2011, 12h00 (Problemas B e C)

Prazo limite de entrega do Relatório: 07/Março/2011, 12h00

Notas Importantes:

- O relatório do trabalho deve ser realizado, MANUSCRITO, no formulário em anexo.
- A não entrega do relatório do trabalho até às 12:00 do dia 07 de Março implica que o aluno tem ZERO na avaliação deste trabalho.

Problemas A, B e C:

Parquímetros Malditos

Os parquímetros da “Estacionamentos & Moedinhas” têm um problema de funcionamento e raramente fornecem aos utentes o troco que é devido.

Foi-lhe pedido que desenvolva um programa eficiente para, sabendo a soma a devolver ao cliente, fazer o cálculo das moedas a entregar. Os pressupostos são os seguintes:

1. A máquina tem moedas em número ilimitado (dos tipos indicados no *input*).
2. O objectivo é **que o troco seja formado pelo menor número possível de moedas**.

Input:

O *input* é composto por duas linhas:

- o valor a devolver ao cliente (1ª linha)
- os valores das moedas disponíveis, por ordem decrescente (2ª linha)

Output:

O *output* deverá compreender um conjunto de linhas, cada linha com um par número de moedas e valor facial da moeda. Estas linhas devem aparecer por ordem decrescente do valor facial das moedas a entregar como troco.

Segue-se uma linha com o número total de moedas. A linha seguinte deve aparecer com um “TROCO EXATO” se é possível fazer o troco exato (formar o conjunto de moedas que perfazem exactamente o troco a entregar) e “SEM TROCO EXATO” se com os tipos de moedas considerados não for possível juntar a soma exata que corresponde ao troco.

Exemplo de *input*

```
11
10 5
```

Exemplo de *output*

```
1 10 [\n]
1 5 [\n]
2 [\n]
SEM TROCO EXATO [\n]
```

Exemplo de *input*

```
24
20 12 1
```

Exemplo de *output*

```
0 20 [\n]
2 12 [\n]
0 1 [\n]
2 [\n]
TROCO EXATO [\n]
```

Notas:

- Quando a solução possível estiver entre devolver menos moedas ou devolver uma soma mais próxima da correta, o programa deve optar pela segunda possibilidade.
- Para somas iguais em excesso o programa deve optar pela que compreenda menos moedas.

Problema A

Construa uma solução para o problema enunciado usando uma solução sôfrega.

Esta solução não tem de produzir necessariamente a melhor solução possível, no sentido de não ter obrigação de minimizar o excesso no troco ou mesmo o número de moedas usadas. A solução poderá produzir, por exemplo, o seguinte resultado (que é incorreto e não seria aceitável para os restantes problemas):

Input

```
39
25 13 5
```

Output

```
1 25[\n]
1 13[\n]
1 5[\n]
3[\n]
SEM TROCO EXATO[\n]
```

Problema B

Construa uma solução para o problema enunciado usando uma solução recursiva.

Esta solução tem obrigação de otimizar o resultado produzido. Por exemplo:

Input

```
39
25 13 5
```

Output

```
0 25[\n]
3 13[\n]
0 5[\n]
3[\n]
TROCO EXATO[\n]
```

Problema C

Construa uma solução para o problema enunciado usando programação dinâmica. Esta solução terá igualmente obrigação de otimizar o resultado produzido.