

## Problema C

# Os trocos trocados

### Problema

Vamos neste exercício exemplificar as diferenças entre programação dinâmica e algoritmos gulosos. Para tal, olhemos para o problema clássico dos trocos em diversos sistemas monetários, alguns esotéricos.

Assumimos aqui que temos moedas ou notas em quantidade suficiente e que o sistema monetário considerado permite a devolução de qualquer troco. Para dar o troco de 32 euros com o menor número de moedas, basta escolher o máximo número de moedas de maior valor abaixo da quantidade por devolver (32) cuja soma não ultrapassa esta quantidade. Aqui trata-se de uma nota de 20 euros. Restando 12 euros por devolver. Aplicamos a mesma receita com o valor 12 e resulta numa nota de 10 euros e, finalmente com os dois euros restantes, uma moeda de 2 euros.

O processo seguido aqui é um clássico algoritmo guloso. Funciona sempre no sistema monetário do Euro. Mas nem todos os sistemas monetários que existiram ou existentes partilham esta benesse<sup>1</sup>.

Vejam os sistemas em que há moedas de 1, 3 e 4. Se tivermos de devolver trocos para o valor 6, o método acima descrito escolherá três moedas : uma moeda de 4 e duas de 1. Mas com duas moedas de 3 teríamos o troco devolvido com menos uma moeda.

É possível desenhar uma solução por programação dinâmica que encontra sempre a melhor forma de dar troco em qualquer sistema monetário.

É a confrontação entre estes dois métodos que vamos aqui exibir neste exercício.

Dados a lista ordenada das moedas de base num sistema monetário, a sua tarefa é dar a indicação se neste sistema monetário a solução ao problema dos trocos por algoritmo guloso é coincidente com o algoritmo por programação dinâmica. Se for idêntica a resposta deverá ser **YES**. Se não for, então deverá devolver a quantidade menor que exibe um comportamento diferenciado nas duas soluções. Por exemplo, no sistema 1,3 e 4, o menor valor em que as duas soluções discordam é precisamente o valor 6 que mencionamos mais acima.

### Entrada

Na primeira linha encontra-se um inteiro  $n$  que indica quantas moedas forma o sistema considerado.

Nas  $n$  linhas seguintes consta um inteiro que representa o valor numérico de cada uma das moedas do sistema monetário considerado.

Estas moedas são apresentadas por ordem crescente e são todas distintas.

### Saída

Uma linha com

- a palavra **YES** caso o sistema monetário em causa permite ao algoritmo guloso concordar com a solução por programação dinâmica;
- ou um inteiro  $p$  que é o montante mais baixo para o qual o troco calcula pelos dois algoritmos é discordante.

### Limites

$0 < n \leq 100$ , e as moedas não ultrapassam o valor 500.

---

<sup>1</sup>Ver para mais detalhes o artigo “Optimal Bounds for the Change Making Problem” de Dexter Kozen e Shmuel Zaks.

### Exemplo de Entrada

3  
1  
3  
4

### Exemplo de Saída

6