# Estruturas de Dados e Algoritmos

Folha de exercícios número 4

2000/2001

1. (Computador e papel) Como sabe o número de combinações de $n$ objectos tomados $m$ a $m$ é

$$\binom{n}{m} = \frac{n!}{m!(n-m)!}$$

   (a) Implemente uma função com o protótipo

```
comb(int n,int m);
```

   que retorna $\binom{n}{m}$, sendo a computação eficiente. Note em particular que a aplicação directa da fórmula indicada é em geral ineficiente (tente calcular à mão $\binom{7}{2}$ usando a fórmula indicada) e comece por derivar uma fórmula mais eficiente. Pode também utilizar a igualdade

$$\binom{n}{m} = \binom{n}{n-m}$$

   seleccionando destas duas opções aquela que envolve menos cálculos.

   O seu programa deve funcionar correctamente para todos os valores positivos de $n$ e $m$.

   (b) Determine, em função de $m$ e $n$, quaantas multiplicações efectua a função que implementou na alínea anterior.

2. (Computador) Considere a representação de um conjunto de elementos, todos compreendidos entre 0 e `MAX-1` por um vector de `MAX` elementos, cada um dos quais só pode conter 0 ou 1, conforme ilustrado para o conjunto $\{1, 2, 7\}$ e para `MAX`$= 10$

$$\{1, 2, 7\} \quad \leftrightarrow \quad \texttt{[0,1,1,0,0,0,0,1,0,0]}$$

   (a) Defina através de `typedef` o tipo `conjunto`.

   (b) Escreva uma instrução cujo efeito é introduzir o elemento `i` no conjunto `s`.

   (c) Escreva uma função com o seguinte protótipo

```
conjunto intersecta(conjunto a, conjunto b);
```

   que retorna a intersecção dos dois conjuntos.

3. (Papel e Computador) Considere um conjunto finito $S$ com $n$ elementos.

   (a) Quantos subconjuntos tem $S$? Justifique.

   (b) Para $0 \leq i \leq n$, quantos subconjuntos de $i$ elementos tem $S$? Justifique.

   (c) Das duas respostas anteriores deduza uma igualdade.

   (d) Seja $S = \{0, 1 \cdots n-1\}$. Implemente um programa que imprima todos os subconjuntos de $S$.

   (e) Seja $S = \{0, 1 \cdots n-1\}$. Implemente um programa que, a partir do valor dado $i$, imprima todos os subconjuntos de $S$ com cardinalidade $i$.

4. (Computador) Um inteiro $n \geq 2$ é primo sse, para todo o $i$ com $1 \leq i \leq n$, for $i^{n-1} = 1 \mod n$. Por exemplo, para $n = 5$ temos

$$\begin{aligned} 1^4 &= 1 = 1 \text{ (mod 5)}, \\ 2^4 &= 16 = 15 + 1 \text{ (mod 5)}, \\ 3^4 &= 81 = 80 + 1 \text{ (mod 5)}, \\ 4^4 &= 246 = 245 + 1 \text{ (mod 5)} \end{aligned}$$

   Se $n$ não é primo, suponha que para a maioria dos $i$ com $1 \leq i \leq n$ temos $i^{n-1} \neq 1 \mod n$. Implemente um programa de computador que testa da seguinte forma se um inteiro dado $n$ é primo:

   Seja $n$ um inteiro fixo. São gerados $m$ inteiros aleatórios $i$ compreendidos entre 2 e $n-1$. Se para qualquer um deles for $i^{n-1} \neq 1 \mod n$, a resposta deve ser NÃO É PRIMO. Caso contrário a resposta deve ser PROVAVELMENTE É PRIMO.

   Nota A potência modular, $a^b \mod n$ deve ser implementada de modo tão eficiente quanto possível e por forma a evitar o aparecimento de inteiros muito grandes.