Estruturas de Dados e Algoritmos

Folha de exercícios número 9

2000/2001

Considere a seguinte função

```
f(n):
  while(n!=1) {
    if(n%2==0) n=n/2 else n=3*n+1;
  }
```

Por exemplo, para o valor inicial `n=7`, temos a seguinte sucessão de 17 valores

```
7, 22, 11, 34, 17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1
```

Pretende-se determinar, dado `m`, o valor (admitindo que só há um) de `n` com $1 \leq$ `n` $\leq$ `m` que corresponde à sequência de comprimento máximo e qual é esse comprimento.
Exemplo Para `m=8` o resultado deve ser `max=17, n=7`.
Nota Apesar da simplicidade da função não se sabe se ela termina para todo o valor inteiro `n`≥ 1.

1. **Simplicidade**
   Implemente um programa tão simples quanto possível com o objectivo indicado.
2. **Tabelação**
   Consegue resolver determinar o comprimento da sequência máxima para `m`$=10^6$? Com vista a solucionar eficientemento o problema para valores grandes de `m`, introduza o seguinte método da *tabelação* no seu programa:
   - Cada vez que um novo valor `v` do comprimento associado a um valor de `n` é calculado, fazemos `comp[n]=v`;
   - o cálculo do comprimento associado a uma sequencia termina quando, para o valor corrente de `n`, o valor de `comp[n]` já está definido.
3. **Desenvolvimento**
   Até que valor de `m` consegue determinar o valor da sequência máxima (e correspondente `n`)? Melhore tanto quanto possível esse limite. As seguintes linhas de acção poderão (ou não) ser úteis ou necessárias:
   - Compilar o programa com maior optimização.
   - Transformar 2 ou mais passos da iteração num só passo.
   - Utilizar aritmética de maior precisão, como por exemplo, `unsigned long int` em vez de `int` (uma transformação simples), dois `int`'s para representar um inteiro ou aritmética de precisão arbitrária. Note que o cálculo do comprimento de uma sequência pode envolver valores intermédios de `n` muito maiores que o valor inicial.