Instituto Superior de Engenharia de Lisboa
Licenciatura em Engenharia Informática e de Computadores
Algoritmos e Estruturas de Dados
Semestre de Inverno 2009/10
Primeira série de exercícios

Observações:

- Data de entrega: **19 de Outubro de 2009**.
- No contexto desta série, o triplo (v, l, r) representa o *subarray* do *array* v, compreendido entre os índices l e r inclusivé.

1 Introdução

1. Realize o método estático

   ```
   public static int removeOdd(int[] v, int count);
   ```

   Este método retira os inteiros ímpares da sequência representada pelos primeiros `count` inteiros do *array* `v`. A sequência resultante fica contida de forma contígua nas primeiras posições do *array* `v`. O valor retornado pelo método é a dimensão da sequência resultante.

2. Realize o método estático

   ```
   public int countEqualTo(int[] v, int l, int r, int a)
   ```

   que retorna o número de ocorrências do inteiro `a` no *subarray*(v, l, r), que está ordenado de forma crescente. O custo assimptótico deve ser $O(\log N)$, onde N é a dimensão do *subarray*.

   Tenha em consideração que o número de ocorrências pertence a $O(N)$.

3. Realize o método estático

   ```
   public static boolean isMaximumSubArrayGivenIndex(int[] v, int l, int r, int i)
   ```

   que retorna *true* se e só se $l \leq i \leq r$ e a soma dos elementos do *subarray* (v, l, r) tiver o maior valor possível para a soma de qualquer *subarray* de v contendo o índice i.

4. Realize o método estático

   ```
   public static IntTriple getMaximumSubArrayGivenIndex(int[] v, int i)
   ```

   que retorna o triplo de inteiros (l, r, s) tal que o resultado da avaliação de `isMaximumSubArrayGivenIndex(v,l,r,i)` seja `true` e $s = \sum_l^r v[i]$. O custo assimptótico deve ser $O(N)$, onde N é a dimensão do *array*.

   Exemplo

   Considere o *array* $v = \{12, -2, -24, 26, -3, -17, -18, 20, 16, -10, 12, -2, 10, -15\}$. O triplo resultante da chamada do método com $i = 3$ é $(3, 12, 34)$. A chamada com $i = 7$ resulta no triplo $(7, 12, 46)$.

5. Considere o método

   ```
   public static IntTriple getMaximumSubArray(int[] v)
   ```

   que retorna o triplo (l, r, s), tal que (v, l, r) seja um *subarray* de v com maior valor para a soma dos seus elementos e s seja o valor dessa soma. Sendo N a dimensão do *array*,

   5.1. Realize uma implementação usando um algoritmo com custo assimptótico $\Theta(N^2)$. Sugestão: pesquisa exaustiva utilizando o método `getMaximumSubArrayGivenIndex`.

   5.2. Realize uma implementação usando um algoritmo com custo assimptótico $\Theta(N \log N)$. Sugestão: divisão do problema em dois subproblemas e utilização do método `getMaximumSubArrayGivenIndex`.

5.3. Realize uma implementação usando um algoritmo com custo assimptótico $\Theta(N)$. Sugestão: percurso linear usando as seguintes observações:

- O primeiro e último elemento do *subarray* solução são não negativos.
- Se o *subarray* (v, l, r) tem soma negativa, então o *subarray* (v, l, r'), para qualquer $r' > r'$, não é solução.

Avalie experimentalmente as implementações anteriores.

2 Análise de desempenho

1. Considere o seguinte algoritmo

```
METHOD(v, l, r, s)
1  if r < l
2      then return s
3  m ← (l + r)/2
4  s ← s + v[m]
5  s ← METHOD(v, l, m − 1, s)
6  s ← METHOD(v, m + 1, r, s)
7  returns;
```

   Indique o número máximo de chamadas ao método METHOD existentes na execução do algoritmo sobre um *sub-array* de dimensão $2^N - 1 = r - l + 1$.

2. Prove que $O(N) + O(N \log N) = O(N \log N)$.

3. Sejam $f, g : \mathbf{N}_0 \longrightarrow \mathbf{R}^+$ funções. Indique se cada uma das seguintes proposições é falsa ou verdadeira.

   3.1. $f = O(g)$ então $g = O(f)$;

   3.2. $f + g = \Theta(min(f, g))$;

   3.3. $f = O(f^2)$;

   Justifique as respostas, apresentando a prova da proposição ou um contra-exemplo.

4. Resolva as seguintes recorrências, usando notação assimptótica para representar o resultado na forma de uma função explícita.

   4.1. $C(n) = C(n/3) + 1$

   4.2. $C(n) = 2C(n/2) + n \log n$.