Exercícios - Análise de Algoritmos Recursivos

Prof. André Vignatti

Exercício 1. Mostre como o `MergeSort` executa no vetor $(3, 41, 52, 26, 38, 57, 9, 49)$.

Exercício 2. Considere o algoritmo Intercala apresentado em aula.

Afirmação (Invariante do Intercala)**.** No começo de cada iteração do laço das linhas 7–12, vale que:

1. $A[p \ldots k-1]$ está ordenado,
2. $A[p \ldots k-1]$ contém todos os elementos de $B[p \ldots i-1]$ e de $B[j+1 \ldots r]$,
3. $B[i] \geq A[k-1]$ e $B[j] \geq A[k-1]$.

(a) Prove que a afirmação acima é de fato um invariante de Intercala.

(b) (fácil) Mostre usando o invariante acima que Intercala é correto.

Exercício 3. Considere o seguinte problema de busca:

Entrada: Uma seqüência de n números $A = (a_1, a_2, \ldots, a_n)$ e um valor v.
Saída: Um índice i tal que $v = A[i]$ ou o valor especial `NIL` se v não aparece em A.

Note que, se o vetor A está ordenado, podemos comparar o elemento $A[n/2]$ com v, eliminando metade do vetor em comparações futuras. Tal procedimento é chamado de `Busca Binária`. Escreva um pseudo-código (recursivo ou iterativo) para a `Busca Binária` e argumente que o tempo execução de pior-caso é $\Theta(\log n)$.

Exercício 4. Observe que o laço interno do `InsertionSort` usa uma busca linear para varrer (de trás para frente) o subvetor ordenado $A[1..j-1]$. Podemos usar a `Busca Binária` (do Exercício 3) para tentar melhorar o pior-caso do `InsertionSort` para $\Theta(n \log n)$?

Exercício 5. O algoritmo abaixo calcula $3^n - 2^n, \forall n \geq 0$. Prove que o algoritmo está correto.

Algoritmo $g(n)$
  se $n \leq 1$ **então retorna** n
  senão retorna $5g(n-1) - 6g(n-2)$

Exercício 6. O algoritmo abaixo calcula a multiplicação de números naturais. Prove que o algoritmo está correto.

Algoritmo $\text{mult}(y, z)$
  se $z = 0$ **então retorna** 0
  senão retorna $\text{mult}(2y, \lfloor z/2 \rfloor) + y(z \bmod 2)$