

Projeto e Análise de Algoritmos

Flávio L. C. de Moura

11 de maio de 2023

1. Sejam $f(n)$, $g(n)$ e $h(n)$ funções não-negativas tais que $f(n) = O(h(n))$ e $g(n) = O(h(n))$. Prove que $f(n) + g(n) = O(h(n))$.
2. Faça a análise assintótica do seguinte algoritmo:

```
1 for i = 0 to n - 2 do
2   for j = n - 1 downto i + 1 do
3     if A[j] < A[j - 1] then
4       swap A[j] and A[j - 1];
5     end
6   end
7 end
```

Algoritmo 1: BubbleSort2($A[0..n - 1]$)

3. Considere o algoritmo a seguir:

```
1 if high < low then
2   return -1;
3 end
4 mid = (high + low) / 2;
5 if key > A[mid] then
6   return BinarySearch(A, mid + 1, high, key);
7 end
8 else
9   if key < A[mid] then
10    return BinarySearch(A, low, mid - 1, key);
11  end
12  else
13    return mid;
14  end
15 end
```

Algoritmo 2: BinarySearch($A[1..n]$, low , $high$, key)

- (a) Faça a análise assintótica do melhor caso;
- (b) Faça a análise assintótica do pior caso;
- (c) Mostre que este algoritmo satisfaz a seguinte propriedade:

Seja $A[1..n]$ um vetor ordenado de inteiros distintos. Mostre que se a chave key não ocorre em $A[1..n]$, então $\text{BinarySearch}(A[1..n], 1, n, key)$ retorna o valor -1.

Dica: Indução (forte) sobre o tamanho n do vetor.