UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO

Bacharelado em Ciência da Computação
Algoritmos e Estruturas de Dados –2022.1

Simulado da 1VA

1. Considere o algoritmo abaixo.

```
1  int minimo (int n, int v[]){
2      int n, m = v[0];    //c1
3
4      for (k=1; k<n; k++){  //c2
5          if (v[k]<m){      //c3
6              m=v[k];       //c4
7          }
8      }
9      return m;             //c5
10 }
```

   a. Encontre uma função de cálculo de custo em função do tamanho do vetor (n) para estimar o tempo de execução. Ex: $T(n) = nc1 + 2nc2 + c3 + \cdots..$

   $T(n) = c1 \cdot 1 + c2 \cdot n + c3 \cdot (n-1) + c4 \cdot (n-1) + c5 \cdot 1$

   T(n) = c1+c2+c3n -c3 +c4n -c4 +c5

   T(n) = (c1+c2-c3-c4+c5) + n(c3+c4)

   T(n) = c'+ n c''

   b. Calcule a complexidade assintótica do algoritmo.

   O(n)

   A complexidade é linear, pois depende apenas do valor do valor de n, à medida que n cresce. Ele cresce de forma linear.

2. Explique através de um passo a passo (com desenho) o funcionamento do InsertionSort. Considere o vetor abaixo.

7	1	8	4	12	9	5	7	9	3

7 | 1 8 4 12 9 5 7 9 3

1 7 | 8 4 12 9 5 7 9 3

1 7 8 |4 12 9 5 7 9 3

1 4 7 8| 12 9 5 7 9 3

1 4 7 8 12 |9 5 7 9 3

1 4 7 8 9 12 | 5 7 9 3

1 4 5 7 8 9 12| 7 9 3

1 4 5 7 7 8 9 12 | 9 3

1 4 5 7 7 8 9 9 12 | 3

1 3 4 5 7 7 8 9 9 12

Qual a complexidade no melhor e pior caso?

Melhor caso: 1 2 3 4| 5 (Array Ordenado)

Melho caso acontece quando vetor está ordenado ou vetor tem apenas elementos iguais.

Número de comparações : n-1 = O(n)

Pior Caso: 5 4 3 2 |1

Número de comparações: 1 + 2 + 3 + 4 + ... (n-1) = n(n-1)/2. = O (n^2)

Pior caso acontece quando o vetor está em ordem decrescente.

3. Implemente o algoritmo de particionamento do QuickSort, fazendo uma modificação no algoritmo de forma que ele considere o pivô sempre o primeiro elemento.

```
int particiona_inicio(A, p, r){
    int x = A[p];
    int i = p;

    for (int j=p+1; j<=r; j++){

        if (A[j]<=x){
            i=i+1;
            troca(A[j], A[i]);
        }

    }
    troca(A[i], A[p]);
    return i;

}
```

4. Ordene os seguintes algoritmos em ordem de complexidade:
   a. O(nlogn)
   b. O(n^2)
   c. O(1000000000)
   d. O(2^n)
   e. O(n^3)
   f. O(log n)
   g. O(n)

c < f < g < a <b < e< d

5. Sobre o Algoritmo SelectionSort:

   a. Descreva e analise uma instância de melhor caso para o algoritmo Selectionsort, ou seja, um vetor v[0..n-1] que leva o algoritmo a executar o menor número possível de comparações.
   Não importa como seja a ordenação inicial do vetor, o selectionsort sempre fará o mesmo número de comparações, que é O(n^2). O algoritmo é independente da ordenação, tendo o mesmo número de comparações para todos os casos.

   b. Quantas vezes, no pior caso, o algoritmo Selectionsort copia um elemento do vetor de um lugar para outro? Quantas vezes isso ocorre no melhor caso?
   O número de trocas é sempre n-1 = O(n). O número de trocas é o mesmo para o melhor ou pior caso, pois ela independe da organização inicial do vetor.