

Lista de Exercícios de Divisão e Conquista

- i) Resolva os problemas abaixo via DC e apresente a equação de recorrência do mesmo.
 - ii) Apresente um pseudo-código para resolver o problema.
 - iii) Apresente a complexidade no pior caso do seu algoritmo.
1. (Análise de Complexidade) Qual a complexidade temporal do seguinte algoritmo?

Algoritmo 0.1 (DC)

Entrada Uma seqüência s_1, \dots, s_n de números inteiros, $n \geq 1$.

```
if (n = 1) then
  return s1
end if
v1 := DC(s1, ..., s[n/2])
v2 := DC(s[n/2]+1, sn)
return min(v1, v2)
```

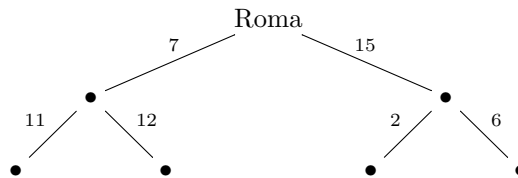
Apresente a equação de recorrência que descreve a complexidade do algoritmo, bem como sua solução.

2. (NewSort) Considere o algoritmo **NewSort()** de ordenação de dados descrito a seguir. O algoritmo **NewSort(S)** recebe como entrada um conjunto S de $|S| = n$ números reais uniformemente distribuídos em R . O algoritmo é um algoritmo recursivo. A cada chamada recursiva o algoritmo
- a) calcula a mediana m dos números da entrada daquela chamada (mediana de um conjunto de n valores é o valor que possui $n/2$ valores menores que ele e $n/2$ valores maiores que ele no conjunto). Este procedimento será chamado de **Med**. O cálculo da mediana de um conjunto de n elementos é feita em tempo $O(n)$.
 - b) divide os números em menores que m (conjunto A), e números maiores ou iguais a m (conjunto B)
 - c) chama recursivamente o algoritmo **NewSort()** duas vezes, uma para o conjunto A e outra para conjunto B
 - d) executa um merge dos dados: o **Merge()** junta o resultado das duas chamadas de forma a manter os dados ordenados.

Se você tivesse que escolher um algoritmo de ordenação, sem informação prévia sobre quais serão os dados de entrada, qual você escolheria entre **NewSort** e **Quicksort**? Porquê? E entre **NewSort** e **MergeSort**? Porquê?

3. (NewSort2) Resolva a questão anterior, substituindo o algoritmo **Med** por um algoritmo que calcula a média aritmética dos valores.

4. (Todos os caminhos levam à Roma) Viajando de Roma para o litoral, descobri que existem várias rotas alternativas com distâncias diferentes que levam para cidades diferentes no litoral. Como a cidade concreta não importa para mim, mas o comprimento do caminho sim, eu gostaria de saber o caminho mais curto de Roma para alguma cidade no litoral. O seguinte exemplo mostra tal situação



com as folhas da árvore sendo as cidades no litoral.

- (a) Qual o caminho mais curto nesse caso?
- (b) Projete um algoritmo, que resolva o problema para árvores binárias completas arbitrárias (enunciado abaixo):

CAMINHO MAIS CURTO EM ÁRVORES

Instância Uma árvore binária completa, com pesos nos links.

Solução O comprimento do caminho mais curto da raiz para alguma folha.

Observação: existe um algoritmo em tempo linear para resolver este problema.

- (c) Justifique a corretude do algoritmo e analise a complexidade dele.
5. Dado um vetor com n números construa um algoritmo de Divisão e Conquista que retorne a posição i em que um dado elemento x ocupa. Suponha que x ocorra no máximo uma vez no vetor.
- a) Escreva o pseudocódigo do algoritmo.
- b) Qual a equação de recorrência do seu algoritmo?
- c) Qual a complexidade do seu algoritmo?
6. Para as equações de recorrência abaixo, apresente sua solução.
- a) $T(n) = 2 \cdot T(\frac{n}{2}) + n \cdot \log n$
- b) $T(n) = 9 \cdot T(\frac{n}{3}) + n$