

Algoritmo dividir e conquistar:

A ordenação em um array pode ser feita de forma exponencial, mas ao dividir o array em blocos, podemos diminuir custo, como o merge sort faz.

Ordenação como no slide 11:

P3: 3, 9, 26, 38, 41, 49, 52, 59

P2: 3, 26, 41, 52 | 9, 38, 49, 59

P1: 3, 41 | 26 52 | 38, 59 | 9, 49

Original : 3, 41, 52, 26, 38, 59, 9, 49

Descrever algoritmo $n \cdot \log n$:

Um algoritmo que ordenaria a lista, depois procuraria valores que somado igual a x;

Formula da recorrência:

Problema 1 :

Se é par: $(a^2) \cdot n/2$

Se é ímpar: $(a^2) (n-1)/2$

Problema 2 (min e max):

Se $n > 2$: $T(n/2) + T(n/2) + 2$

Se $n = 2$: 1

Se $n = 1$: 0

Exemplo divisão e conquista em programação:

Código de a^n em python:

```
def potencia(num, power):
```

```
    if power == 0:
```

```
        return 1
```

```
    elif power == 1:
```

```
        return num
```

```
    elif power & 1 == 0:
```

```
        return potenciacao(num * num, power // 2)
```

```
    elif power & 1 == 1:
```

```
        return num * potenciacao(num * num, (power - 1) // 2)
```

```
num, power = map(int, input("Digite dois números em sequência: ").split())
```

```
print(potencia(num, power))
```