Técnicas de Programação

Divisão e Conquista

Ordenação até agora

- Bubble Sort
- Selection Sort

Bubble Sort

```
BUBBLE_SORT(A)
    N ← TAMANHO(A)
    FEZ_TROCA ← VERDADEIRO

    ENQUANTO FEZ_TROCA FAÇA
        FEZ_TROCA ← FALSO
        PARA CADA I ← 0 ATÉ N-1 FAÇA
            SE A[I] > A[I+1] ENTÃO
                TROQUE OS VALORES DE A[I] E A[I+1]
                FEZ_TROCA ← VERDADEIRO
            FIM
        FIM
    FIM
```

Bubble Sort

Características:

- melhor caso => loop de fora dá uma volta, de dentro faz `N` iterações
- pior caso => loop de fora dá `N` voltas, de dentro faz `N` iterações

Pior caso é N^2!

Selection Sort

```
SELECTION_SORT(A)
    N ← TAMANHO(A)
    PARA I = 0 ATÉ N - 1 FAÇA
        MENOR_IDX = I
        PARA J = I+1 ATÉ N FAÇA
            SE A[J] < A[MENOR_IDX] FAÇA
                MENOR_IDX = J
            FIM
        FIM
        TROCA VALORES DE A[I] E A[MENOR_IDX]
    FIM
```

Selection Sort

Características:

- sempre roda igual => loop externo roda `N-1` vezes, loop interno roda `N-I` vezes para o `i` atual

Tempo final também é N^2!

Selection Sort vs Bubble Sort

- Se já estiver ordenado (ou quase isso), Bubble é melhor
- Selection faz menos escritas no vetor e isso é mais rápido em geral
- No pior caso ambos são iguais, mas o array que atinge o pior caso é diferente para ambos

Divisão e Conquista

Idéia 1: Quick Sort

1. escolha um elemento do vetor (pode ser o primeiro)
2. coloque ele no lugar certo
    - índice `i` tal que todo $A[j] \leq A[i], j < i$
    - índice `i` tal que todo $A[j] \geq A[i], j > i$
3. ordene a parte à esquerda de `i`
4. ordene a parte à direita de `i`

Prática 1

Atividade Quick Sort

Idéia 2: Merge Sort

1. divida o vetor em duas metades
2. ordene a metade da direita
3. ordene a metade da esquerda
4. mescle as duas metades ordenadas tal que agora o vetor inteiro esteja ordenado

Prática 2

Atividade Merge Sort

Divisão e Conquista

1. dividir os dados em partes
2. executar o algoritmo em cada parte
3. combinar os resultados das partes para o vetor inteiro

APS

Implementações dos algoritmos da aula de hoje e testes grandes do problema