



ALGORITMOS E PROGRAMAÇÃO II

AULA 06 – Algoritmos de Ordenação: Selection Sort

Prof. Thomás da Costa
<https://thomasdacosta.com.br/>

Acessibilidade

- Foi criado um descritivo inicial em cada slide que explica se o slide possui imagem ou não e qual a relevância na disciplina;
- Caso tenha uma imagem relevante, o texto inicial explica o conteúdo da imagem;
- Nos slides de código fonte existe um texto inicial explicando como o código fonte está organizado e se o mesmo está completo ou é somente um único trecho de código para explicar o conceito da disciplina;
- Sempre que aparecer o texto #DescriçãoDoSlide representa o texto explicativo do slide;
- Estarei aberto a sugestões para melhorar a acessibilidade do material de aula para todos os alunos.

Selection Sort

- O algoritmo tem como objetivo encontrar o menor ou maior valor e incluir na posição mais a esquerda;
- É composto por dois laços, o primeiro laço tem como objetivo percorrer o vetor e o segundo encontrar o menor número em relação a posição atual do vetor;



#ordenação



#DescriçãoDoSlide: Código fonte desse slide é somente um trecho para explicar o conceito da disciplina. O código não está completo.

Código fonte completo no link: <https://github.com/thomasdacosta/senac-java-algoritmos-programacao-2/blob/main/src/main/java/edu/senac/aula08/SelectionSort.java>

```
public static void main(String[] args) throws IOException {
    Integer valores[] = new Integer[TAMANHO_VETOR];

    Random random = new Random();

    for (int i=0;i<=TAMANHO_VETOR-1;i++)
        valores[i] = random.ints(1, TAMANHO_VETOR).findAny().getAsInt();

    int posAtual = 0;
    int posMinimoValor = 0;
    boolean troca = false;
    for (int i=0;i<=valores.length-1;i++) {
        posAtual = i;
        posMinimoValor = i;
        troca = false;
        for (int j=i;j<=valores.length-1;j++) {
            if (valores[posMinimoValor] > valores[j]) {
                posMinimoValor = j;
                troca = true;
            }
        }

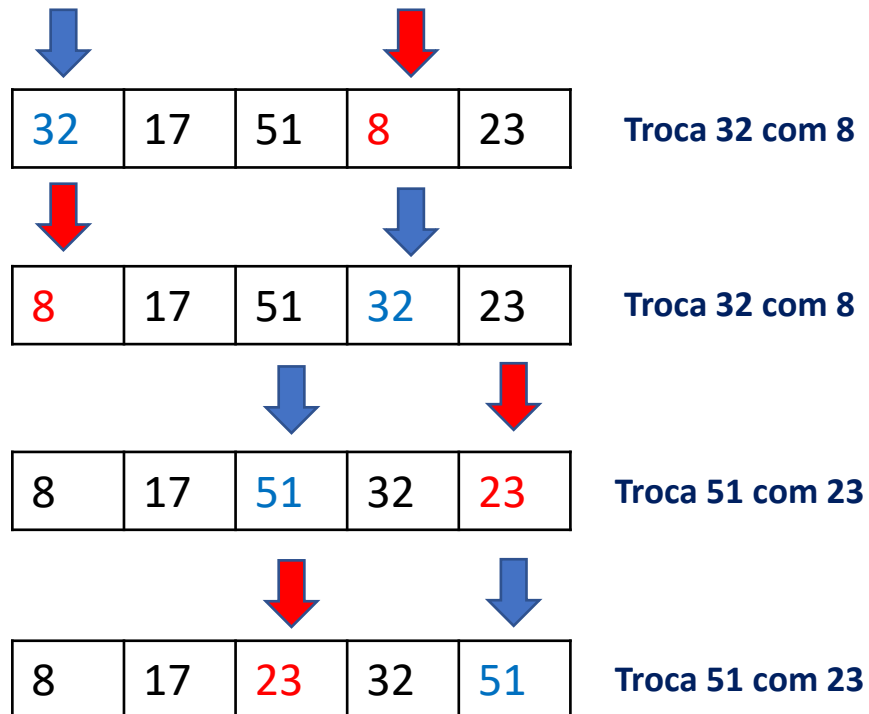
        if (troca) {
            int valorAtual = valores[posAtual];
            int valorMinimoValor = valores[posMinimoValor];

            valores[posAtual] = valorMinimoValor;
            valores[posMinimoValor] = valorAtual;
        }
    }
}
```

#ordenação

#DescriçãoDoSlide: Código fonte desse slide é somente um trecho para explicar o conceito da disciplina. O código não está completo.

Selection Sort



Vetor ordenado corretamente!!!



Bubble Sort vs Selection Sort

- Vetor 1K elementos:
 - Bubble Sort: 0.0120099S
 - Selection Sort: 0.0040326S
- Vetor 10K elementos:
 - Bubble Sort: 0.3192311S
 - Selection Sort: 0.0872364S
- Vetor 100K elementos:
 - Bubble Sort: 38.2941481S
 - Selection Sort: 11.3109703S

Selection Sort é 66.422701271451% mais rápido



Selection Sort é 72.67296325452% mais rápido



Selection Sort é 70.462927390368% mais rápido



Fontes

- <https://embarcados.com.br/algoritmos-de-ordenacao-bubble-sort/>
- <https://www.devmedia.com.br/algoritmos-de-ordenacao-analise-e-comparacao/28261>
- <https://ic.unicamp.br/~mc102/aulas/aula10.pdf>
- <https://ic.unicamp.br/~mc102/aulas/aula11.pdf>
- <https://www.geeksforgeeks.org/java-program-for-selection-sort/#:~:text=The%20selection%20sort%20algorithm%20sorts,subarray%20which%20is%20already%20sorted.>





Obrigado.