

Tópicos Avançados em Estruturas de Dados

Atividade 6

Bruna Galastri Guedes	18.00189-0
Daniel Ughini Xavier	18.00022-3
Rodolfo Cochi Bezerra	18.00202-0
Vítor Martin Simoni	18.00050-9
Leonardo de Barros Rodrigues	18.02401-7

22/04/2020

Questão 1

O algoritmo a ser analisado, "selection sort", é o seguinte:

```
#include <stdio.h>
int main()
{
    int array[100], n, c, d, position, t;

    printf("Enter number of elements\n");
    scanf("%d", &n);

    printf("Enter %d integers\n", n);

    for (c = 0; c < n; c++)
        scanf("%d", &array[c]);

    for (c = 0; c < (n - 1); c++) // finding minimum element (n-1) times
    {
        position = c;

        for (d = c + 1; d < n; d++)
        {
            if (array[position] > array[d])
                position = d;
        }
        if (position != c)
        {
            t = array[c];
            array[c] = array[position];
            array[position] = t;
        }
    }

    printf("Sorted list in ascending order:\n");

    for (c = 0; c < n; c++)
        printf("%d\n", array[c]);

    return 0;
}
```

Analisando o algoritmo "selection sort", percebe-se que este compara um elemento com outro a cada iteração, visando encontrar o menor dentre eles; assim, é possível entender que não existe melhor caso, visto que não importa a ordem de ordenação do vetor, já que os dois laços do algoritmo (externo e interno) serão sempre executados. Sendo assim, conclui-se que:

- a) A ordem de complexidade do melhor caso é $O(n^2)$.
- b) A ordem de complexidade do pior caso é $O(n^2)$.