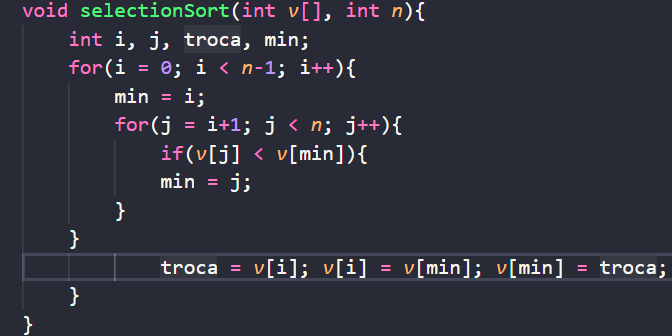
Selection Sort



Primeiro vamos a uma breve explicação de como funciona o Selection Sort ele é um algoritmo baseado para sempre passar o menor valor do vetor para a primeira posição exemplo.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 01 | 02 | 03 | 04 | 05 | 06 |
| **3** | **1** | **9** | **5** | **2** | **8** |

Na tabela acima ele verifica qual o menor numero da tabela no caso é o número 1 com isso ele realiza uma troca com o número que está na posição 1 da tabela como podemos ver na tabela abaixo.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 01 | 02 | 03 | 04 | 05 | 06 |
| **1** | **3** | **9** | **5** | **2** | **8** |

E assim que ele realizou a troca ele deixa de verificar o numero 1 pois ele já está na posição correta para começar a verificar as demais posições, novamente ele segue procurando o menor número da tabela para realizar a troca com o numero que se encontra na posição 2 da tabela e assim consecutivamente até chegarmos a uma tabela totalmente ordenada.

Agora vamos a explicação do código acima feito em (C), podemos criar uma função que retorna o índice do menor elemento de um vetor vet (formado por n números inteiros) a partir de uma posição inicial dada no caso o i=0 e notamos também que não é necessário o nosso n ir até o ultimo número ele pode ir somente até o penúltimo que fica i = n -1.

Abaixo definimos como realizar a troca.

Troca= v[i]; v[i] = v[min]; v[min] = troca;

Nesta linha apenas trocamos o valor de v[i] pelo v[min] definindo assim que o v[min] deve ser trocado com o numero que estava na posição v[i].

Complexidade

Considerando um array com N elementos, o tempo de execução é sempre de ordem O(N2 ) pois a eficiência do selection sort não depende da ordem inicial dos elementos.