A ideia por tráz do insertion sort é achar um local para colocar um elemento, ou seja, vc vai mudar outros elementos de lugar para inserir o seu elemento.

int insertionsort(int \*v, int n){

for (int i = 1; i < n; i++){

int j = i - 1;

int t = v[i]; //achar a posicao para t

while (j >= 0 && v[j] > t){

v[j + 1] = v[j];

j--;

}

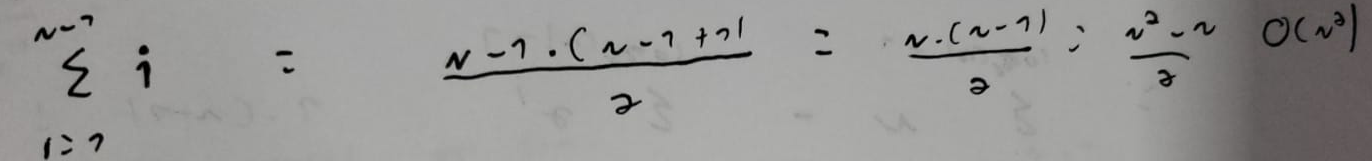
v[j + 1] = t;

}

}

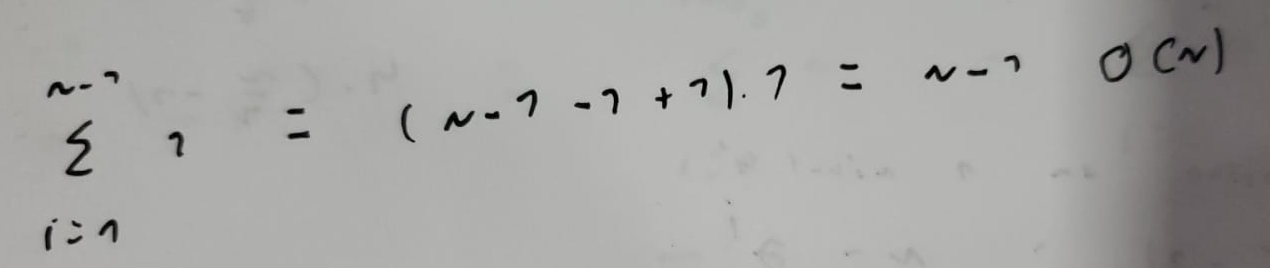
Pior caso:  
Array totalmente desordenado

Como o array está totalmente desordenado sabemos que o while vai rodar i vezes em cada iteração do i, logo, será:



Melhor caso:  
Arrayo totalmente ordenado

Como o array está totalmente ordenado, a cada iteração o while só vai rodar 1 vez (checando q ja n precisa fazer trocas), logo será:



Isso é só para as comparações, tb tem para o número de atribuições de variáveis, por exemplo, que leva a exatamente a mesma lógica, mas levar em consideração essas atribuições não vai mudar o big O.

Ela só é boa em arquivos quase ordenados.