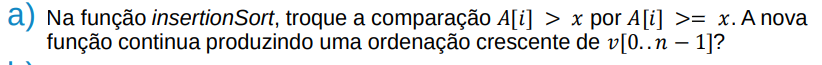
1. Considerando o algoritmo de ordenação por inserção:   
   

Sim



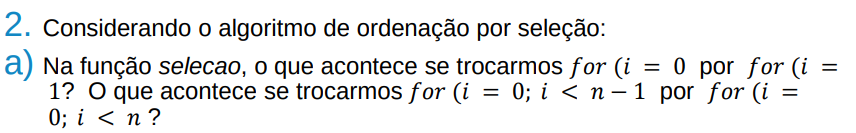
Continua ordenando, mas o primeiro passo, o primeiro elemento é inserido na sua própria posição.



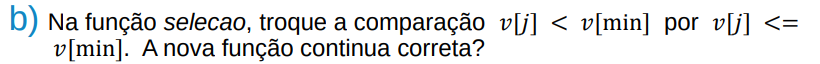
Irá sobrescrecer registros. Imagine que tenhamos o seguinte vetor [2,3,1]. Na primeira execução j=1, A[j]=x= 3 e i=0. No condicional o i é >=0, mas A[i]=2 não é maior que x=3. Então A[i] receberá 3 resultando no seguinte vetor [3,3,1]



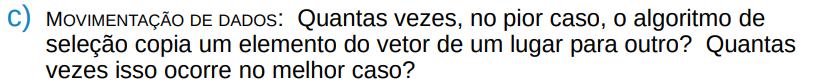
Basta apenas alterar a condição de teste de valor para (A[i] < x)



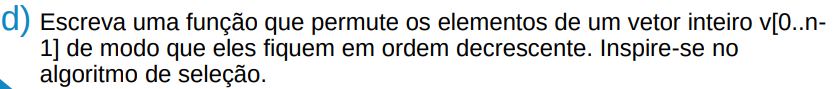
A primeira troca deixa de ordenar o primeiro elemento. Já a segunda troca só realizará uma execução a mais do loop trocando o último elemento com ele mesmo.



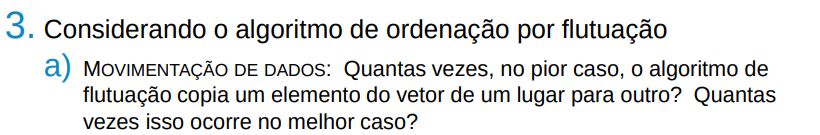
Continuará correto pois apenas irá trocar posições de valores iguais.



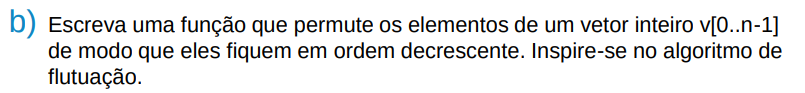
a troca é constante. Da primeira a última interação há uma troca, totalizando n-1 trocas na execução completa.



Basta realizar a inversão do teste de valores para if (v[j] > v[min])



No pior caso há n2-1 e no melhor caso não há nenuma troca.



Basta alterar o condicional de valor para if (v[j] < v[j + 1])

CÓDIGO

#include <limits.h>

#include <stdio.h>

#include <stdlib.h>

#define tamanho 5

void bubble(int v[tamanho], int n)

{

int i, j, aux;

int k = n - 1;

for (i = 0; i < n; i++)

{

for (j = 0; j < k; j++)

{

if (v[j] > v[j + 1])

{

aux = v[j];

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

v[j + 1] = aux;

}

}

k--;

}

}

void bubbleDesc(int v[tamanho], int n)

{

int i, j, aux;

int k = n - 1;

for (i = 0; i < n; i++)

{

for (j = 0; j < k; j++)

{

if (v[j] < v[j + 1])

{

aux = v[j];

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

v[j + 1] = aux;

}

}

k--;

}

}

void selecao(int v[tamanho], int n)

{

int i, j, min, x;

for (i = 0; i < n - 1; ++i)

{

min = i;

for (j = i + 1; j < n; ++j)

if (v[j] < v[min])

min = j;

x = v[i];

v[i] = v[min];

v[min] = x;

}

}

void selecaoDesc(int v[tamanho], int n)

{

int i, j, min, x;

for (i = 0; i < n - 1; ++i)

{

min = i;

for (j = i + 1; j < n; ++j)

if (v[j] > v[min])

min = j;

x = v[i];

v[i] = v[min];

v[min] = x;

}

}

void insertionSort(int A[tamanho], int n)

{

int i, j, x, count = 0;

for (j = 1; j < n; j++)

{

x = A[j];

i = j - 1;

while ((i >= 0) && (A[i] > x))

{

A[i + 1] = A[i];

i = i - 1;

count++;

}

A[i + 1] = x;

}

printf("troca: %d \n", count);

}

void insertionSortDesc(int A[tamanho], int n)

{

int i, j, x;

for (j = 1; j < n; j++)

{

x = A[j];

i = j - 1;

while ((i >= 0) && (A[i] < x))

{

A[i + 1] = A[i];

i = i - 1;

}

A[i + 1] = x;

}

}

int main()

{

int vetor[tamanho];

vetor[0] = 3;

vetor[1] = 1;

vetor[2] = 5;

vetor[3] = 9;

vetor[4] = 7;

bubbleDesc(vetor, tamanho);

for (int i = 0; i < tamanho; i++)

{

printf("%d ", vetor[i]);

}

printf("\n");

return 0;

}