

Passar exemplo com alunos: 8 – 7 – 10 – 15 – 12 – 4

Dar uma lista de valores e pedir para alunos simularem o caso

9 – 7 – 10 – 1 – 5 – 20 – 15

9 e 7 – troca – 7 – 9 – 10 – 1 – 5 – 20 – 15

9 e 10 – nada

10 e 1 – troca – 7 – 9 – 1 – 10 – 5 – 20 – 15

10 e 5 – troca – 7 – 9 – 1 – 5 – 10 – 20 – 15

10 e 20 – não troca

20 e 15 – troca - 7 – 9 – 1 – 5 – 10 – 15 – 20

7 e 9 – não troca

9 e 1 – troca - 7 – 1 – 9 – 5 – 10 – 15 – 20

9 e 5 – troca - 7 – 1 – 5 – 9 – 10 – 15 – 20

9 e 10 – não troca

10 e 15 – não troca - 7 – 1 – 5 – 9 – 10 – 15 – 20

7 e 1 - troca - 1 – 7 – 5 – 9 – 10 – 15 – 20

7 e 5 – troca - 1 – 5 – 7 – 9 – 10 – 15 – 20

7 e 9 – não troca

9 e 10 – não troca - 1 – 5 – 7 – 9 – 10 – 15 – 20

1 e 5 – não troca

5 e 7 – não troca

7 e 9 – não troca - 1 – 5 – 7 – 9 – 10 – 15 – 20

1 e 5 – não troca

5 e 7 – não troca - 1 – 5 – 7 – 9 – 10 – 15 – 20

1 e 5 – não troca - 1 – 5 – 7 – 9 – 10 – 15 – 20

Algoritmo do Bubble Sort

|  |  |
| --- | --- |
| Bubble(tot: inteiro; ent[]: inteiro)  i,j,aux: inteiro;  para i = 1 até tot faça  para j = 1 até tot-i faça  se ent[j+1] < ent[j] entao  aux 🡨 ent[j];  ent[j] 🡨 ent[j+1];  ent[j+1] 🡨 aux;  fim\_se  fim\_para  fim\_para  fim | Bubble(tot: inteiro; ent[]: inteiro)  i,j,aux, troca: inteiro;  troca 🡨 1;  para i = 1 até tot faça  se troca=1 entao  troca 🡨 0;  para j = 1 até tot-i faça  se ent[j+1] < ent[j] entao  aux 🡨 ent[j];  ent[j] 🡨 ent[j+1];  ent[j+1] 🡨 aux;  troca 🡨 1;  fim\_se  fim\_para  fim\_se  fim\_para  fim |

**EXPLICAR – QUANDO NÃO TROCAR, PODE PARAR – JÁ ORDENADO – PEQUENA MELHORIA**

Programa C

|  |  |
| --- | --- |
| #include <stdio.h>  #include <stdlib.h>  void ordenar(int ent[],int tot)  {  int i,j,aux;  for (i=1;i<tot;i++)  {  for (j=1;j<=tot-i;j++)  {  if (ent[j+1] < ent[j])  {  aux = ent[j];  ent[j]=ent[j+1];  ent[j+1] = aux;  }  }  }  } | **com melhoria (var troca)**  void ordenar(int ent[],int tot)  {  int i,j,aux,*contarpassos*,troca;  *contarpassos=0;*  troca=1;  for (i=1;i<tot;i++)  {  if (troca==1)  {  troca=0;  for (j=1;j<=tot-i;j++)  {  *contarpassos++;*  if (ent[j+1] < ent[j])  {  aux = ent[j];  ent[j]=ent[j+1];  ent[j+1] = aux;  troca=1;  }  }  }  }  *printf("%d\n",contarpassos);*  } |

**Pedir para alunos simularem caso com parada para os valores:**

1 – 5 – 10 – 7 – 8 – 9 – somente 2 passos

Lista já ordenada: 1 – 7 - 9 – 11 – somente 1 passo

Número de passos é variável (pressupondo existência da variável troca)

Número de comparações em cada passo não é variável – n-1 comparações

**Mostrar que O(n2) – sem melhoria. Três casos iguais (melhor, pior, caso médio)**

Analisar bubble sobre os 3 aspectos

Em termos de comparação (com melhoria)

Melhor caso com parada O(n) = n-1 – JÁ ORDENADO

Caso médio ou pior caso O(n2) = n(n-1)/2 – é provável que pare antes, 2/3.

Pior caso – ordem inversa: troca sempre. O(n2) = n(n-1)/2

**Atentar que o numero de trocas pode ser grande.**

**Método estável**

Mostrar quadro resumo

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | SELEÇÃO | INSERÇÃO | BOLHA (Bubble) – sem melhoria | BOLHA (Bubble) – com melhoria (troca) |
| Estavel x Instavel | Instavel | Estavel | Estavel | Estavel |
| Melhor caso | O(n2) | O(n) - ordenado | O(n2) | O(n) - ordenado |
| Caso Médio | O(n2) | O(n2) | O(n2) | O(n2) |
| Pior Caso | O(n2) | O(n2) – inversamente ordenado | O(n2) | O(n2) – inversamente ordenado |