

Método de ordenação Bolha  
Prof. Tiago Eugenio de Melo  
[tiagodemelo@gmail.com](mailto:tiagodemelo@gmail.com)

# Bolha

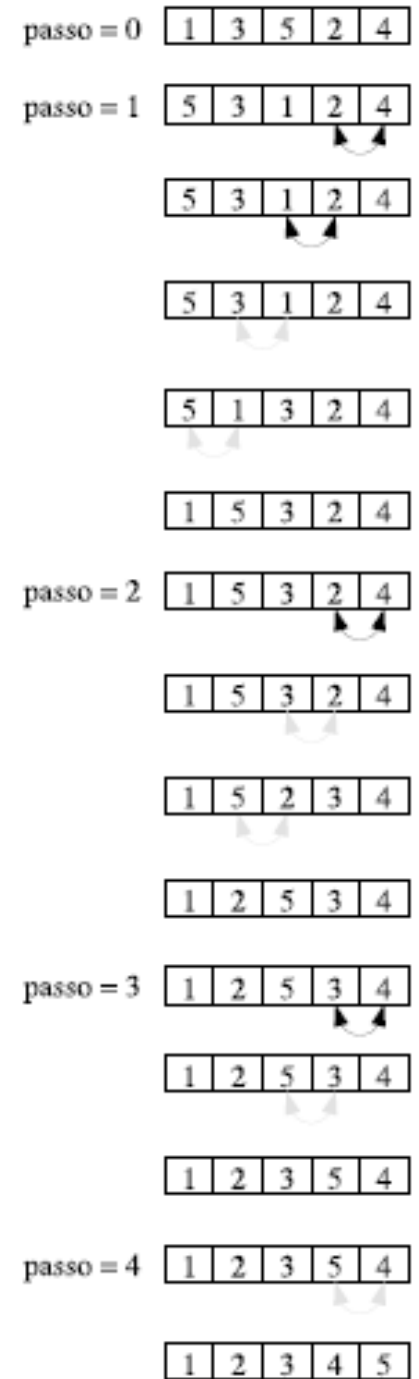
- O método mais simples e, talvez, o mais conhecido método de ordenação por troca.
-

# Bolha

- Idéia
  - Compara elementos adjacentes. Se o segundo for menor, realiza a troca.
  - Esta operação é repetida desde do primeiro até o último par.

# Bolha

- Exemplo



# Bolha

- Implementação

```
1 void Bolha (int A[], int n) {
2     int i, j, temp;
3
4     for (i = 1; i < n; i++){
5         for (j = n; j > i; j--) {
6             if (A[j-1] > A[j]) {
7                 temp = A[j-1];
8                 A[j-1] = A[j];
9                 A[j] = temp;
10            }
11        }
12    }
12 }
```

# Bolha

- É possível melhorar a implementação anterior, evitando as passagens desnecessárias, através da identificação de quando a entrada já está ordenada.
- Se numa passagem, não houver nenhuma troca, a entrada estará ordenada.
- Para isto, basta controlar a ocorrência de trocas.

# Bolha

- Implementação melhorada

```
bubble (x, n)
int x[], n;
{
    int hold, j, pass;
    int switched = TRUE;

    for (pass = 0; pass < n-1 && switched == TRUE; pass++) {
        /* repeticao mais externa controla nº de passagens */
        switched = FALSE; /* inicialmente nenhuma troca */
                           /* foi feita nesta passagem */
        for (j = 0; j < n-pass-1; j++)
            /* repeticao mais interna controla cada pass indiv */
            if (x[j] > x[j+1]) {
                /* elementos fora de ordem */
                /* eh necessaria uma troca */
                switched = TRUE;
                hold = x[j];
                x[j] = x[j+1];
                x[j+1] = hold;
            } /* fim if */
        } /* fim for */
    } /* fim bubble */
```

# Análise do algoritmo

- O algoritmo depende da seqüência inicial da entrada.
- O algoritmo utiliza duas estruturas de repetição aninhada.
- Grosseiramente, no laço interno são executadas, no máximo,  $N - 1$  comparações.
- Pior caso
  - $(N - 1) * (N - 1)$  comparações, portanto é  $O(n^2)$ .

# Análise do algoritmo

- Vantagens:
  - Simplicidade.
  - Exigência de pouco espaço adicional (um registro extra para armazenar o valor temporário da troca e diversas variáveis inteiras simples).
  - O fato de ser  $O(n)$  no caso em que o arquivo está totalmente classificado.
    - Isto se verifica ao observar que apenas uma passagem de  $n - 1$  comparações (e nenhuma troca) é necessária para perceber que a entrada já está ordenada.