ALGORITMOS DE ORDENAÇÃO SIMPLES

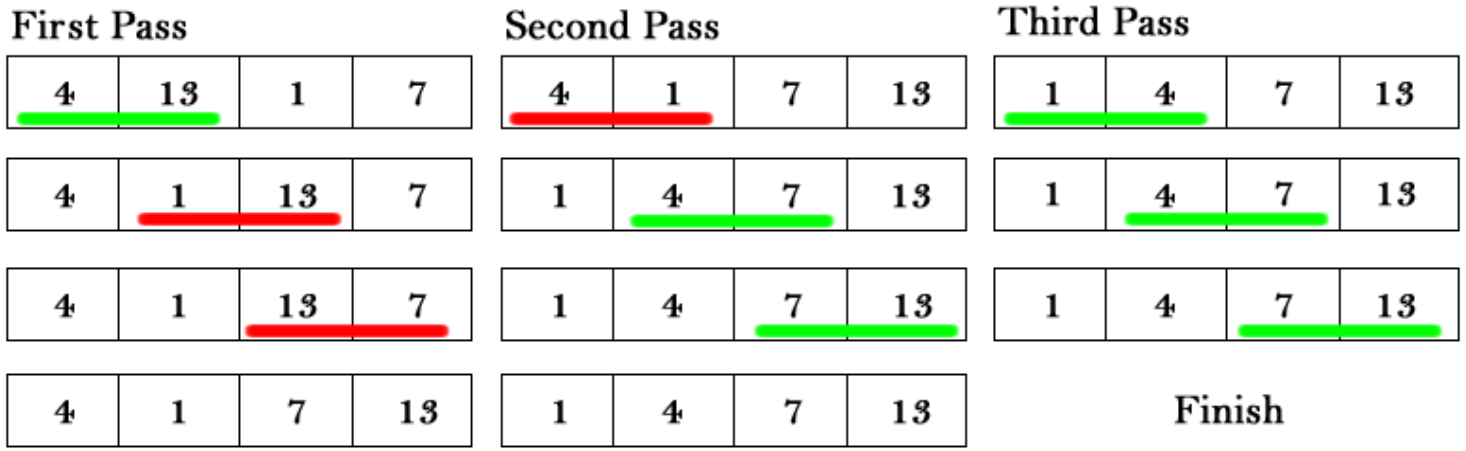
Prof. Me. Hélio Esperidião

Bubble sort

O bubble sort, ou ordenação por flutuação (literalmente "por bolha"), é um algoritmo simples.

A idéia é percorrer o vetor diversas vezes até que ele esteja ordenado, a cada passagem o maior elemento é deslocado (flutua) para o final do vetor

Bubble sort



- Essa movimentação lembra a forma como as bolhas em um tanque de água procuram seu próprio nível, e disso vem o nome do algoritmo.

Bubble sort

- Faça uma análise do Bubble sort quanto a sua complexidade temporal e espacial.

Selection sort

- Compara o elemento atual com todos os elementos do vetor se existir algum menor as posições são trocadas.

- Tem como principio sempre passar o menor valor do vetor para a primeira posição (ou o maior dependendo da ordem requerida), depois o de segundo menor valor para a segunda posição, e assim é feito sucessivamente com os (n-1) elementos restantes.

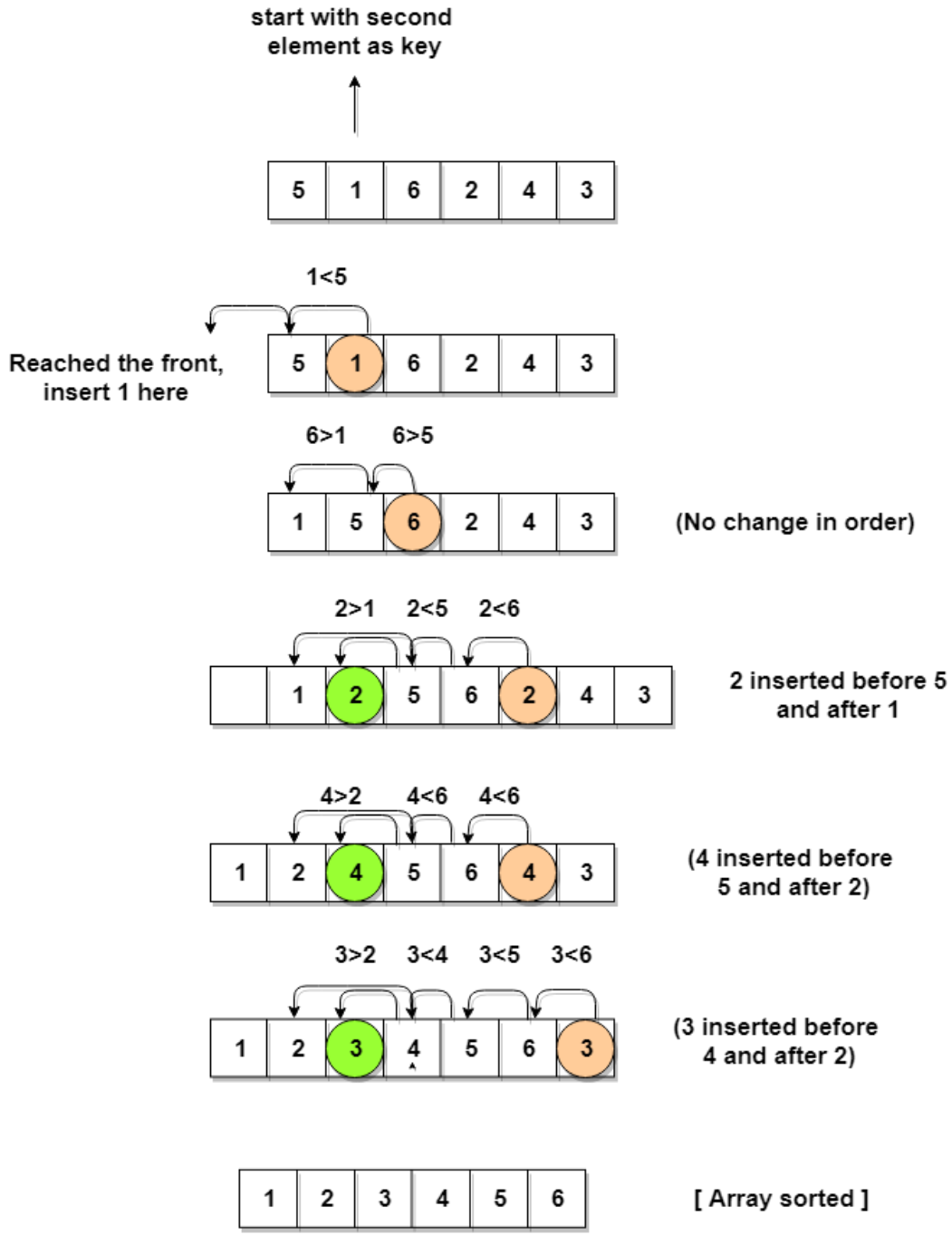
Selection sort

Find Smallest	21	73	32	10 ↑	42
Swap with First	10	73	32	21	42
Find Smallest	10	73	32	21 ↑	42
Swap with First	10	21	32	73	42
Find Smallest	10	21	32 ↑	73	42
Swap with First	10	21	32	73	42
Find Smallest	10	21	32	73	42 ↑
Swap with First	10	21	32	42	73

Insertion sort

6 5 3 1 8 7 2 4

- Ele percorre um vetor de elementos da esquerda para a direita e à medida que avança vai deixando os elementos mais à esquerda ordenados.


start with second element as key
5 1 6 2 4 3
1<5
Reached the front, insert 1 here
5 1 6 2 4 3
6>1 6>5
1 5 6 2 4 3
(No change in order)
2>1 2<5 2<6
1 2 5 6 2 4 3
2 inserted before 5 and after 1
4>2 4<6 4<6
1 2 4 5 6 4 3
(4 inserted before 5 and after 2)
3>2 3<4 3<5 3<6
1 2 3 4 5 6 3
(3 inserted before 4 and after 2)
1 2 3 4 5 6
[Array sorted]

Insertion sort

- Faça uma análise quanto a complexidade espacial e temporal.

Exercícios.

- Implementar os três algoritmos de ordenação nas linguagens c, java e C#.
- Criar e executar uma estratégia para testar o desempenho na ordenação em cada uma das linguagens.
- Crie um programa em c que seja capaz de ler um texto e colocar todas as letras do texto em ordem alfabética.
- Faça testes com estruturas grandes na ordem de milhões de elementos.
  - Armazene os tempos de execução e verifique se possuem de fato desempenho quadrático.

- Analise o código e determine o que ele faz.
- Implemente o insertion sort utilizando ponteiros.
- Pesquise e implemente o selection sort utilizando ponteiros .

```c
int main(){
   int *p;
   int qt, n, i, j, aux;
   printf("\nQuantas rodadas ? ");
   scanf("%d",&qt);
   while (qt != 0)   {
      printf("\nQuantos numeros ? ");
      scanf("%d", &n);
      p = (int*)calloc(n, sizeof(int));

      // printf("Digites %d numeros e tecle ENTER: ", n);
      for (i = 0; i < n; i++)      {
         printf("%d/%d: ", i+1, n);
         scanf("%d", (p+i));
      }
      for (i = 0; i < n-1; i++)      {
         for (j = i+1; j < n; j++)         {
            if (p[i] > p[j])            {
               aux = p[i];
               p[i] = p[j];
               p[j] = aux;
            } // if
         } // for j
      } // for i

      for (i = 0; i < n; i++)      {
         printf("%d ", p[i]);
      }
      printf("\n");
      free(p);
      qt--;

   } // while
}
```