Algoritmos de Ordenação - Selection Sort

SCC0201 - Introdução à Ciência de Computação II

Clausius G. Reis Leandro A. Amaral Tiago S. Nazaré
Vanessa Q. Marinho

Instituto de Ciências Matemáticas e de Computação
Universidade de São Paulo

21 de junho de 2015

Agenda

Agenda

Introdução

É um dos algoritmos mais simples de ordenação.

Ideia Básica:

- **Ordem Crescente**: Selecione o menor elemento e troque-o com o item da primeira posição do vetor. Repita essas operações com os $n-1$ elementos restantes, depois com os $n-2$ elementos, até que reste apenas um elemento.
- **Ordem Decrescente**: Selecione o maior elemento e troque-o com o item da primeira posição do vetor. Repita essas operações com os $n-1$ elementos restantes, depois com os $n-2$ elementos, até que reste apenas um elemento.

Agenda

Exemplo - Ordem Crescente

5	6	3	4	1
5	6	3	4	1
1	6	3	4	5
1	**6**	3	4	5
1	3	6	4	5
1	3	**6**	4	5
1	3	4	6	5
1	3	4	**6**	5
1	3	4	5	6

Agenda

Selection Sort - Código

```
void selection_sort(int *a, int size){
        int i, j, min, aux;

        for(i=0; i<size-1; ++i){
                min = i;

                for(j= i+1; j < size; ++j){
                        if(a[j] < a[min])
                                min = j;
                }

                aux = a[min];
                a[min] = a[i];
                a[i] = aux;
        }
}
```

Agenda

Análise de Complexidade

Para ordenar um vetor com n elementos:

Complexidade: $\mathcal{O}(n^2)$ - **Melhor, pior e caso médio**

- Para encontrar o menor elemento é necessário acessar os n elementos (realizando $n-1$ comparações) e trocar com a primeira posição;
- Para encontrar o segundo menor elemento é necessário acessar os $n-1$ elementos restantes e assim por diante. Portanto:
  $(n-1)+(n-2)+\ldots+2+1 = n(n-1)/2 = \mathcal{O}(n^2)$

Obs.: O fato do vetor já estar ordenado não ajuda, pois o custo continua quadrático.

Exercício

Implementar (em C) o algoritmo de Selection Sort de maneira recursiva. **Dica:** Considere o início do vetor como variável.

Resposta

```
void selection_sort_rec ( int *a, int init, int size) {
    int min, aux, i;
    min = init;
    for (i = init; i < size; i++) {
        if (a[i] < a[min]) {
            min = i;
        }
    }

    aux = a[min];
    a[min] = a[init];
    a[init] = aux;

    if (init < size) {
        selection_sort_rec (a, ++init, size) ;
    }
}
void main() {
    selection_sort_rec(a, 0, size);
}
```