Notas de Aula - AED1 – Algoritmos de Ordenação (parte 3)
Prof. Jefferson T. Oliva

Hoje veremos os dois últimos algoritmos de ordenação antes da segunda prova.

A ordenação por inserção classifica um conjunto de registros inserindo registros em um arquivo classificado existente.

Exemplos de algoritmos de ordenação por inserção: insertion sort e shell sort.

Insertion sort

É o método que consiste em inserir informações num conjunto já ordenado.

Algoritmo utilizado pelo jogador de cartas
- As cartas são mantidas ordenadas nas mãos dos jogadores
- Quando o jogador compra ou recebe uma nova carta, ele procura a posição que a nova carta deverá ocupar

O algoritmo é um dos mais eficientes para pequenas entradas.

Implementação

```
void insertsort(int v[], int n){
        int i , x;

        for (i = 1; i < n; i++){
                x = v[i];

                for (j = i - 1; (j >= 0) && (x  < v[j]); j--)
                        v[j + 1] = v[j];

                v[j + 1] = x;
        }
}
```

Ver slides 6 e 7

Desempenho do método
- Melhor caso: O(n)
- Médio caso: O(n^2)
- Pior caso: O(n^2)

Deve ser utilizado quando o arquivo está "quase" ordenado.
Desvantagem: alto custo de movimentação de elementos.

Ver slide 9.

Variações do Método:
=> inserção com pesquisa binária: consiste em utilizar o método da busca binária para localizar a posição a ser inserido o elemento:
→ Diminui o número de comparações mas ainda é necessário efetuar o deslocamento dos elementos para a inserção.
→ Isso sendo executado n vezes resulta em O(n^2) substituições.
→De modo geral não ajuda!
=> Inserção em lista ligada: consiste em não mover as informações e sim efetuar as inserções nas ligações. Portanto, o tempo gasto com comparações continua sendo O(n^2),
=> A melhor variação é a inserção com incrementos decrescentes, também chamado de ordenação de Shell (shell sort)

Shell sort

Extensão do insert sort

Contorna o principal problema do insertion sort possibilitando troca de registros que estão distantes um do outro.

Tem como objetivo aumentar o passo de movimento dos elementos ao invés das posições adjacentes.

Consiste em classificar sub-arranjos do original.

Esses sub-arranjos contêm todo h-ésimo elemento do arranjo original.

O valor de h é chamado de incremento.

Por exemplo, se h é 5, o sub-arranjo consiste dos elementos x[0], x[5], x[10], etc
- Sub-arranjo 1: x[0], x[5], x[10]
- Sub-arranjo 2: x[1], x[6], x[11]
- Sub-arranjo 3: x[2], x[7], x[12]
- Sub-arranjo 4: x[3], x[8], x[13]

Após a ordenação dos sub-arranjos:
- Define-se um novo incremento menor que o anterior
- Gera-se novos sub-arquivos
- Aplica-se novamente o método da inserção

O processo é realizado repetidamente até que h seja igual a 1

O valor de h pode ser definido de várias formas, por exemplo
- h(s) = 3h(s − 1) + 1, para s > 1
- h(s) = 1, para s = 1

Implementação

```
void shellsort(int v[], int n){
        int h = 1;
        int x, i, j;

        while (h < n)
                h - 3 * h + 1;

        h /= 3;

        while (h >= 1){
                for (i = h; i < n; i++){
                        x = v[i];
                        j = i;

                        while ((j >= h) && (x < v[j - h])){
                                v[j] = v[j - h];
                                j -= h;
                        }

                        v[j] = x;
                }

                h /= 3;
        }
}
```

Ver exemplo no slide 14.

Um problema com o shell sort ainda não resolvido é a escolha dos incrementos que fornecem os melhores resultados.

É desejável que ocorra o maior número possível de interações entre as diversas cadeias.

Ótima opção para arquivos de tamanho moderado.

Tempo de execução sensível à ordem dos dados.

O algoritmo não é estável.
Custo de tempo não é conhecido, mas estima-se aproximadamente:

- Pior caso: O(n^2)
- Melhor caso: O(n ^{1,25})

ver slide 17

Referências

Cormen, T. H., Leiserson, C. E., Rivest, R. L., Stein, C. Introduction to Algorithms. Third edition, The MIT Press, 2009.

Horowitz, E., Sahni, S. Rajasekaran, S. Computer Algorithms. Computer Science Press, 1998.

Rosa, J. L. G. Métodos de Ordenação. SCE-181 - Introdução à Ciência da Computação II. Slides. Ciência de Computação. ICMC/USP, 2018.

Ziviani, N. Projeto de Algoritmos - com implementações em Java e C++. Thomson, 2007.