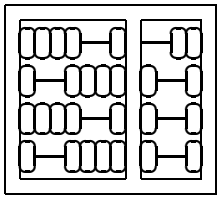

MC-102 Algoritmos e Programação de Computadores
IC-UNICAMP

Aula 17B - Aula sobre métodos de ordenação

1 Objetivos

Apresentar diferentes métodos de ordenação, destacando suas diferenças e as vantagens e desvantagens de cada um.

2 Motivação

- Mostrar os métodos de ordenação, seus pontos fracos e também onde cada método é mais adequado, permitindo assim que a escolha do método possa atender a aplicação.

3 Métodos de Ordenação

O problema da ordenação é fundamental na computação, como vimos nos métodos de pesquisa a eficiência da busca sempre é melhor quando trabalhamos com conjuntos ordenados.

Existem muitos algoritmos de ordenação, a escolha do mais eficiente vai depender de vários fatores: número de itens a ser classificado; se os valores já estão agrupados em subconjuntos ordenados; etc.

3.1 Ordenação por Inserção

A proposta da ordenação por inserção é:

```
Para cada elemento do vetor faça
    insira-o na sua posição correspondente;
```

Durante o processo de ordenação por inserção a lista fica dividida em duas sub listas, uma com os elementos já ordenados e a outra com elementos ainda por ordenar.

No início, a sub lista ordenada é formada trivialmente apenas pelo primeiro elemento da lista.

Nos métodos de ordenação por inserção, a cada etapa i, o i-ésimo elemento é inserido em seu lugar apropriado entre os(i-1) elementos já ordenados. Os índices dos itens a serem inseridos variam 2 a m.

Veja no exemplo abaixo o resultado das varreduras:

Varredura	X[0]	X[1]	X[2]	X[3]	X[4]
Vetor original	9	8	7	6	5
1	{8	9}	{7	6	5}
2	{7	8	9}	{6	5}
3	{6	7	8	9}	{5}
4	{5	6	7	8	9}

3.1.1 Função de Inserção

```
void Insercao(int m,int vet[])
{
   int i,j,aux;
   for(i = 1; i < m; i++)
    {
      aux = vet[i];
      for(j = i-1; (j >= 0) && (aux < vet[j]); j--)
          vet[j + 1] = vet[j];
          vet[j + 1] = aux;
    }
}
```

Número de Comparações:

Em cada etapa i são executadas no máximo, i comparações pois o loop interno é executado para j de i-1 até 0. Como i varia de 1 ate m temos, pela formula da soma dos termos de uma P.A.:

$\sum_{i=1}^{m} a_i = (1+m) * \frac{(m+1)}{2} = O(m^2)$

3.2 Ordenação por troca - *Bubble Sort*

Um método simples de ordenação por troca é a estratégia conhecida como bolha que consiste, em cada etapa "borbulhar" o maior elemento para o fim da lista. Inicialmente percorre-se a lista dada da esquerda para a direita, comparando pares de elementos consecutivos, trocando de lugar os que estão fora da ordem.No exemplo abaixo, em cada troca, o maior elemento é deslocado uma posição para a direita.

Varredura	X[0]	X[1]	X[2]	X[3]	Troca
1	10	9	7	6	0 e 1
	9	10	7	6	1 e 2
	9	7	10	6	2 e 3
	9	7	6	10	Fim da Varredura 1

Após a primeira varredura o maior elemento encontra-se alocado em sua posição definitiva na lista ordenada. Podemos deixá-lo de lado e efetuar a segunda varredura na sub lista v[0],v[1],v[2]. Veja a continuação do exemplo:

Varredura	X[0]	X[1]	X[2]	X[3]	Troca
2	9	7	6	10	0 e 1
	7	9	6	10	1 e 2
	7	6	9	10	Fim da Varredura 2

Após a segunda varredura o maior elemento da sub lista v[0],v[1],v[2] encontra-se alocado em sua posição definitiva. A próxima sub lista a ser ordenada é v[0],v[1]. Veja a continuação do exemplo:

Varredura	X[0]	X[1]	X[2]	X[3]	Troca
3	7	6	9	10	0 e 1
	6	7	6	10	Fim da Varredura 3

3.2.1 Função Bubble Sort

```
void BubbleSort(int m,int x[])
{
  int aux,j,i;
  for(i=0;i<m-1;i++)
  {
    for(j=0;j<m-i-1;j++)
      if (x[j] > x[j+1]) {
        aux=x[j];
        x[j]=x[j+1];
        x[j+1]=aux;
      }
  }
}
```

Número de Comparações:

No algoritmo da bolha observamos que existem m-1 varreduras(etapas) e que em cada varredura o número de de comparações diminui de uma unidade , variando de m-1 até 1. Portanto :

$\sum_{i=1}^{m-1}(m-1) = (m-1) + (m-2) + ... + 1 = \frac{m(m-1)}{2} = O(m^2)$

3.3 Exercícios

1. Ordene o vetor v=20,12,28,05,10,18 usando o método de inserção. Mostre o vetor a cada passo do *loop*.

   R.

	V[0]	**V[1]**	**V[2]**	**V[3]**	**V[4]**	**V[5]**	i
V	20	12	28	05	10	18	0
	20						1
	12	20					2
	12	20	28				3
	05	12	20	28			4
	05	10	12	20	28		5
	05	10	12	18	20	28	6

2. Ordene o vetor v=20,12,28,05,10,18 usando o método de bolha. Mostre o vetor a cada passo do *loop*.

   R.

	V[0]	**V[1]**	**V[2]**	**V[3]**	**V[4]**	**V[5]**	i
V	20	12	28	05	10	18	
1 Passo	12	20	05	28			
				10	28		
					28	28	
2 Passo	12	05	10	18	20	28	
3 Passo	05	10	12	18	20	28	
4 Passo							

3. Ordene o vetor v=20,12,28,05,10,18 usando a função abaixo. Mostre o vetor a cada passo do *loop*. Após explique o algoritmo.

   R.

```
void Ordena(int m,int x[])
{
  int aux,j,i;
  for(i=m-1;i>=0;i--){
    for(j=m-1;j>(m-1)-i;j--)
      if (x[j-1] > x[j]) {
        aux=x[j];
        x[j]=x[j-1];
        x[j-1]=aux;
      }
  }
}
```

	V[0]	**V[1]**	**V[2]**	**V[3]**	**V[4]**	**V[5]**	i
V	20	12	28	05	10	18	
1 Passo	05	20	12	28	10	18	
	05	10	20	12	28	18	
	05	10	12	20	18	28	
2 Passo	05	10	12	18	20	28	
3 Passo	05	10	12	18	20	28	
4 Passo	05	10	12	18	20	28	

   É a função *BubbleSort* ao contrário. Ou seja, percorre o vetor da direita para esquerda e vai empurrando os menores para o início.

4. Refaça a função de ordenação por seleção, mas agora buscando o menor elemento e não o maior como foi mostrado. Após Ordene o vetor v=20,12,28,05,10,18 usando a função criada. Mostre o vetor a cada passo do *loop*.

   R.

```
void Selecao(int m,int x[]) /* Buscando o menor */
{
  int aux,j,i,menor;
  for(i=0;i<m-1;i++)
  {
    menor=i+1;
    for(j=i+1;j<m;j++)
      if (x[j] < x[menor])
          menor=j;
    aux=x[i];
    x[i]=x[menor];
    x[menor]=aux;
    mostra(m,x);
  }
}
```

	V[0]	**V[1]**	**V[2]**	**V[3]**	**V[4]**	**V[5]**	i
V	20	12	28	05	10	18	0
	05	12	28	20	10	18	1
	05	10	28	20	12	18	2
	05	10	12	20	28	18	3
	05	10	12	18	28	20	4
	05	10	12	18	20	28	5

4 Bibliografia Utilizada

Apostila do **prof. Flávio Keidi Miyazawa**.

Paulo Feofiloff - Projeto de Algoritmos. disponível em: http://www.ime.usp.br/ pf/algoritmos

Fernando Lobo. Programação Imperativa, 2003/04. Disponível em :http://www.adeec.fct.ualg.pt/PI_flobo/

Material disponível em: http://www.inf.pucpcaldas.br/biblioteca_virtual/c/metodos_de_ordenacao.htm

Material do Curso de C da UFMG - Disponível em:http://ead1.eee.ufmg.br/cursos/C/