

## Ordenação Parcial

### 1. Algoritmos de Ordenação Parcial [Ziviani, 2011; Capítulo 4, Seção 4.1.6]

#### a) Seleção Parcial

- i) Altere o algoritmo Ordenação por Seleção apresentado no Programa 1 para obter somente os  $k$  primeiros itens em ordem ascendente. Liste o número e o conteúdo das linhas modificadas.
- ii) Obtenha as funções  $C(n)$  e  $M(n)$ , as quais denotam o número de comparações entre chaves de registros e movimentações de registros, respectivamente.

#### b) Inserção Parcial

- i) Repita o item a) para o algoritmo de Ordenação por Inserção apresentado no Programa 2.

#### c) Heapsort Parcial

- i) Repita o item a) para o algoritmo do Heapsort apresentado no Programa 3.

#### d) Quicksort Parcial

- i) Repita o item a) para o algoritmo do Quicksort apresentado no Programa 4.

#### Programa 1: Algoritmo de Ordenação por Seleção

```
1. void Selecao(TipoItem *A, TipoIndice n)
2. { TipoIndice i, j, Min;
3.   TipoItem x;
4.   for (i = 1; i <= n - 1; i++)
5.     { Min = i;
6.       for (j = i + 1; j <= n; j++)
7.         if (A[j].Chave < A[Min].Chave) Min = j;
8.       x = A[Min]; A[Min] = A[i]; A[i] = x;
9.     }
10. }
```

#### Programa 2: Algoritmo de Ordenação por Inserção

```
1. void Insercao(TipoItem *A, TipoIndice n)
2. { TipoIndice i, j;
3.   TipoItem x;
4.   for (i = 2; i <= n; i++)
5.     { x = A[i]; j = i - 1;
6.       A[0] = x; /* sentinela */
7.       while (x.Chave < A[j].Chave)
8.         { A[j+1] = A[j]; j--;
9.       }
10.      A[j+1] = x;
11.    }
12. }
```

#### Programa 3: Algoritmo do Heapsort

```
1. void Refaz(TipoIndice Esq, TipoIndice Dir, TipoItem *A)
2. { TipoIndice i = Esq;
3.   int j;
4.   TipoItem x;
5.   j = i * 2;
6.   x = A[i];
7.   while (j <= Dir)
8.     { if (j < Dir)
9.       { if (A[j].Chave < A[j+1].Chave)
10.        j++;
11.      }
12.     }
```

```

12.     if (x.Chave >= A[j].Chave) goto L999;
13.     A[i] = A[j];
14.     i = j;  j = i * 2;
15.     }
16. L999: A[i] = x;
17. }
18.
19. void Constroi(TipoItem *A, TipoIndice n)
20. { TipoIndice Esq;
21.   Esq = n / 2 + 1;
22.   while (Esq > 1)
23.     { Esq--;
24.       Refaz(Esq, n, A);
25.     }
26. }
27.
28. void Heapsort(TipoItem *A, TipoIndice n)
29. { TipoIndice Esq, Dir;
30.   TipoItem x;
31.   Constroi(A, n); /* constroi o heap */
32.   Esq = 1;  Dir = n;
33.   while (Dir > 1)
34.     { /* ordena o vetor */
35.       x = A[1];  A[1] = A[Dir];  A[Dir] = x;  Dir--;
36.       Refaz(Esq, Dir, A);
37.     }
38. }

```

#### Programa 4: Algoritmo do Quicksort

```

1. void Particao(TipoIndice Esq, TipoIndice Dir,
2.             TipoIndice *i, TipoIndice *j, TipoItem *A)
3. { TipoItem x, w;
4.   *i = Esq;  *j = Dir;
5.   x = A[(*i + *j) / 2]; /* obtem o pivo x */
6.   do
7.     { while (x.Chave > A[*i].Chave) (*i)++;
8.       while (x.Chave < A[*j].Chave) (*j)--;
9.       if (*i <= *j)
10.        { w = A[*i]; A[*i] = A[*j]; A[*j] = w;
11.          (*i)++; (*j)--;
12.        }
13.     } while (*i <= *j);
14. }
15.
16. void Ordena(TipoIndice Esq, TipoIndice Dir, TipoItem *A)
17. { TipoIndice i, j;
18.   Particao(Esq, Dir, &i, &j, A);
19.   if (Esq < j) Ordena(Esq, j, A);
20.   if (i < Dir) Ordena(i, Dir, A);
21. }
22.
23. void QuickSort(TipoItem *A, TipoIndice n)
24. { Ordena(1, n, A); }

```

#### Referência

N. Ziviani. *Projeto de Algoritmos: com Implementações em PASCAL e C*. 3a. edição revista e ampliada. São Paulo: CENGAGE Learning, 2011.