unipac.br
Barbacena

Bacharelado em Ciência da Computação

Estruturas de Dados

Material de Apoio

Parte XII – *Método QuickSort*

Prof. Nairon Neri Silva
naironsilva@unipac.br

2º sem / 2021

Ordenação por Quicksort

- Proposto por Hoare em 1960 e publicado em 1962.
- É o algoritmo de ordenação interna mais rápido que se conhece para uma ampla variedade de situações.
- Provavelmente é o mais utilizado.
- A ideia básica é dividir o problema de ordenar um conjunto com n itens em dois problemas menores.
- Os problemas menores são ordenados independentemente.
- Os resultados são combinados para produzir a solução final.

Ordenação por Quicksort

- A parte mais delicada do método é o processo de partição.
- O vetor A [Esq..Dir] é rearranjado por meio da escolha arbitrária de um **pivô** x.
- O vetor A é particionado em duas partes:
  - Parte esquerda: chaves $\leq$ x.
  - Parte direita: chaves $\geq$ x.

- **O algoritmo apresentado define o pivô como o elemento mais a esquerda dos dados.**

Ordenação por Quicksort

- Função para o particionamento:
  1. O método “particao” separa os dados entre pivô, conjunto de dados menores que o pivô e conjunto de dados maiores que o pivô.
  2. Ao final, retorna a posição final do pivô, que já está na ordem correta.

Ordenação por Quicksort - Algoritmo

```
int particao(int v[], int esq, int dir){
     int pivo, i, j, aux;
     pivo = v[esq];
     i = esq;
     j = dir;
     while(i<j){
        while((v[i] <= pivo) && (i <= dir))
            i++;
        while(v[j] > pivo){
            j--;
        if(i<j){
            aux = v[i];
            v[i] = v[j];
            v[j] = aux;
        }
     }
     v[esq] = v[j];
     v[j] = pivo;
     return j;
}
```

Ordenação por Quicksort

- Ao final do algoritmo de partição, o vetor A[Esq..Dir] está particionado de tal forma que:
  - Os itens em A[Esq], A[Esq + 1], ..., A[j] são menores ou iguais ao pivô;
  - Os itens em A[j], A[j + 1], ..., A[Dir] são maiores ou iguais ao pivô.
- Então, o método QuickSort faz chamadas recursivas para a metade esquerda (menor ou igual ao pivô) e a metade direita (maior ou igual ao pivô).

Ordenação por Quicksort - Algoritmo

```
void QuickSort(int v[], int esq, int dir) {
     int i;

     if (dir > esq){
        pivo = particao(v, esq, dir);
        quickSort(v,esq,pivo-1); //metade esquerda
        quickSort(v,pivo+1,dir); //metade direita
     }
}
```

Ordenação por Quicksort

- O pivô x é escolhido como sendo o elemento mais a esquerda do conjunto de dados.

- Exemplo:

3	6	4	5	1	7	2

- O pivô será o 3 (primeiro elemento)

3	6	4	5	1	7	2

1	2	3	5	4	7	6

1	2	3	5	4	7	6

1	2	3	5	4	7	6

1	2	3	4	5	7	6

1	2	3	4	5	7	6

1	2	3	4	5	6	7

1	2	3	4	5	6	7

Ordenação por Quicksort

- O anel interno da função “particao” é extremamente simples.

- Razão pela qual o algoritmo Quicksort é tão rápido.

Ordenação por QuickSort

- **Características**

  – Qual o pior caso para o Quicksort?
    - Por que?
    - Qual sua ordem de complexidade?

  – Qual o melhor caso?

  – O algoritmo é estável?

Algoritmo Quicksort

- **Análise**
  - Pior caso:
    - O pior caso ocorre quando, sistematicamente, o pivô é escolhido como sendo um dos extremos de um arquivo já ordenado.
    - Isto faz com que o procedimento seja chamado recursivamente n vezes, eliminando apenas um item em cada chamada.
    - O pior caso pode ser evitado empregando pequenas modificações no algoritmo.
    - Para isso basta escolher três itens quaisquer do vetor e usar a **mediana dos três** como pivô.

Algoritmo Quicksort

- Vantagens:
  - É extremamente eficiente para ordenar arquivos de dados.
  - Necessita de apenas uma pequena pilha como memória auxiliar.

- Desvantagens:
  - Sua implementação é muito delicada e difícil:
    - Um pequeno engano pode levar a efeitos inesperados para algumas entradas de dados.
  - O método não é **estável**.

QuickSort - Exercício

Ordene o seguinte conjunto de dados usando o algoritmo QuickSort:

V = {75, 63, 7, 84, 3, 2, 4, 0, 9}

Referências

- FORBELLONE, André Luiz Villar; EBERSPACHER, Henri Frederico. *Lógica de Programação*. Makron books.
- GUIMARAES, Angelo de Moura; LAGES, Newton Alberto Castilho. *Algoritmos e estruturas de dados*. LTC Editora.
- FIDALGO, Robson. *Material para aulas*. UFRPE.

- NELSON, Fábio. *Material para aulas: Algoritmo e Programação.* UNIVASP.
- FEOFILOFF, P., *Algoritmos em linguagem C*, Editora Campus, 2008.
- ZIVIANI, N., *Projeto de algoritmos com Implementações em Pascal e C*, São Paulo: Pioneira, 2d, 2004.
- http://www.ime.usp.br/~pf/algoritmos/
- MELLO, Ronaldo S., *Material para aulas: Ordenação de Dados*, UFSC-CTC-INE
- MENOTTI, David, *Material para aulas: Algoritmos e Estrutura de Dados I*, DECOM-UFOP