# cid:image001.jpg@01CE5BA8.2704C360

Cristian Karcezewski

**ALGORITIMOS DE ORDENAÇÃO**

**Orientador: Prof. Sandra Rovena Frigieri**

**Disciplina: Estrutura de Dados**

Caxias do Sul

2015 / 2

**SUMÁRIO**

|  |  |
| --- | --- |
| INTRODUÇÃO | 03 |
| 1. O QUE É O QUICKSORT | 04 |
| 2. LÓGICA DE FUNCIONAMENTO | 04 |
| 3.IMPLEMENTAÇÃO E UTILIZAÇÃO DO QUICKSORT | 04 |
| 4. PERFORMACE, ANALISE E EFICIENCIA DO MÉTODO | 06 |
| CONCLUSÃO | 07 |
| BIBLIOGRAFIA | 08 |

**INTRODUÇÃO**

O presente trabalho é sobre o algoritmo de ordenação chamado quicksort, mais concretamente uma pequena pesquisa sobre o que é o quicksort a sua lógica de funcionamento e como ele é implantado.

Os objetivos deste trabalho é explicar brevemente os fatores citado acima. Está organizado em quatro partes, são elas: O que é o quicksort; Lógica de funcionamento; Implementação e utilização do quicksort e por fim a performance, analise e comparação com os outros métodos de ordenação.

A metodologia utilizada foi a pesquisa via internet e também algumas informações retiradas da plataforma moodle da Ftec Faculdades.

**1. O QUE É O QUICKSORT**

O algoritmo Quicksort é o método de ordenação mais rápido e eficiente, foi inventado por Charles Antony Richard Hoare em 1960, na tentativa de traduzir um dicionário em inglês para o russo. O quicksort trabalha com o método de divisão e conquista, por este motivo é bastante eficiente.

**2. LÓGICA DE FUNCIONAMENTO**

Basicamente, o quicksort vai dividindo o vetor ao meio recursivamente, até que não seja mais possível dividir e depois retorna ordenando. Por ser recursivo, necessita de uma pequena quantidade de memória adicional, o principal cuidado a ser tomado é a escolha de onde dividir o vetor. Caso ele trabalhe com um vetor já ordenado ele vai demorar mais tempo do que com vetor não ordenado.

**3. IMPLEMENTAÇÃO E UTILIZAÇÃO DO QUICKSORT**

O quicksort pode ser implementado em qualquer programa que necessita ordenar qualquer tipo de dado. É muito utilizado principalmente com banco de dados.

//Main

int main()

{

int n=10, vetor[10] = { 3 , 5 , 8 , 1 , 9 , 2 , 4 , 7 , 0 , 6 };

quicksort( vetor , 0 , n );

for( int i = 0 ; i < 10 ; i++ ){

printf( "%d - ", vetor[i] );

}

return 0;

}

//estrutura básica do quick sort

void quicksort( int vet[], int esq, int dir ){

int pivo;

if( esq < dir ){

pivo = divide( vet , esq , dir );

quicksort( vet, esq , pivo-1 );

quicksort( vet , pivo+1 , dir );

}

}

//função de troca de inteiros

void troca( int \*y , int \*x ){

int aux;

aux = \*x;

\*x = \*y;

\*y = aux;

}

//função que divide o vetor

int divide( int vet[] , int esq , int dir ){

int count = esq;

for( int i = esq+1 ; i <= dir ; i++){

if( vet[i] < vet[esq] ){

count += 1;

troca( &vet[i] , &vet[count] );

}

}

troca ( &vet[esq] , &vet[count] );

return count;

**}**

**4. PERFORMACE, ANALISE E EFICIENCIA DO MÉTODO**

O método que levou menos tempo real de execução recebeu o valor de 1 e os outros receberam tempos relativos.

* Elementos em ordem aleatória:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 5.00 | 5.000 | 10.000 | 30.000 |
| Inserção | 11,3 | 87 | 161 | - |
| Seleção | 16,2 | 124 | 228 | - |
| Shellsort | 1,2 | 1,6 | 1,7 | 2 |
| Quicksort | 1 | 1 | 1 | 1 |
| Heapsort | 1,5 | 1,6 | 1,6 | 1,6 |

* Elementos em ordem crescente:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 5.00 | 5.000 | 10.000 | 30.000 |
| Inserção | 1 | 1 | 1 | 1 |
| Seleção | 128 | 1.524 | 3.066 | - |
| Shellsort | 3,9 | 6,8 | 7,3 | 8,1 |
| Quicksort | 4,1 | 6,3 | 6,8 | 7,1 |
| Heapsort | 12,2 | 20,8 | 22,4 | 24,6 |

* Elementos em ordem decrescente:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 5.00 | 5.000 | 10.000 | 30.000 |
| Inserção | 40,3 | 305 | 575 | - |
| Seleção | 29,3 | 221 | 417 | - |
| Shellsort | 1,5 | 1,5 | 1,6 | 1,6 |
| Quicksort | 1 | 1 | 1 | 1 |
| Heapsort | 2,5 | 2,7 | 2,7 | 2,9 |

**CONCLUSÃO**

Neste trabalho foi abordado o assunto quicksort, que é um dos algoritmos mais usados para ordenação dentro de programas e banco de dados. Foi concluído que comparando com os outros métodos de ordenação, é um dos métodos mais rápidos e eficiente na maioria das situações.

**BIBLIOGRAFIA**

http://homepages.dcc.ufmg.br/~cunha/teaching/20121/aeds2/sortingcmp.pdf

https://www.youtube.com/watch?v=iT5Iofq6i3U

http://homepages.dcc.ufmg.br/~cunha/teaching/20121/aeds2/quicksort.pdf