




Algoritmo de Ordenação Externa:

QuickSort Externo

Base de dados

| DRS | Município | Unidade de Saúde | Nomes de Medida | Leitos Ocupados / UTI |
|------------|---------------------|---|------------------------------|-----------------------|
| Araçatuba | ARAÇATUBA | HOSPITAL MUNICIPAL DA MULHER DR JOAO LUIS J ROSSETO | Letios Ocupados / UTI | 1 |
| Araçatuba | ARAÇATUBA | HOSPITAL MUNICIPAL DA MULHER DR JOAO LUIS J ROSSETO | Leitos Ocupados / Enfermaria | 0 |
| Araraquara | AMÉRICO BRASILIENSE | UNIDADE HOSPITALAR DR. JOSÉ NIGRO NETO | Leticos Ocupados / UTI | 0 |
| Araraquara | AMÉRICO BRASILIENSE | HOSPITAL ESTADUAL AMÉRICO BRASILIENSE | Leitos Ocupados / Enfermaria | 3 |
| Araraquara | AMÉRICO BRASILIENSE | UNIDADE HOSPITALAR DR. JOSÉ NIGRO NETO | Leticos Ocupados / UTI | 3 |



Como funciona o QuickSort

- Algoritmo de Divisão e Conquista
- Ordenação realizada *in situ*
- Complexidade $O(n \log n)$
- Pior caso: $O(n^2)$ (extremamente raro)
- Método de particionamento
- Como escolher o pivô

[Simule o QuickSort](#)

Algoritmo de Ordenação Externa

O que é e para
que serve?

- Método de ordenação para arquivos muito grandes
- Se utiliza quando não é possível alocar em memória primária
- Quebra o arquivo em blocos





Como funciona o QuickSort Externo

- Semelhante ao QuickSort interno
- Particionamento
- Chamadas recursivas



Algoritmos em C



Funções auxiliares

```
1 void lerSup(FILE *arqIS, void *reg, int *IS, int nBytes, short int *ordem);
2
3 void lerInf(FILE *arqII, void *reg, int *II, int nBytes, short int *ordem);
4
5 void escSup(FILE *arqES, void *reg, int *eS, int nBytes);
6
7 void escInf(FILE *arqEI, void *reg, int *eI, int nBytes);
8
9 void inserirArea(void *area, int *areaOcupada, void *reg, int nBytes, int(*comp)(void*,void*));
10
11 void rmvPrimeiroArea(void *area, int *areaOcupada, int nBytes);
12
13 void particao(FILE *arq,int inicio, int fim, int *i, int *j, int tamArea, int nBytes, int(*comp)(void*,void*));
14
15 int comparaCrescente(void *a, void *b);
16
17 int comparaDecrescente(void *a, void *b);
18
19 void quickSortExterno(FILE *arq, int inicio, int fim, int tamArea, int nBytes, int (comp)(void*,void*));
20
```

Funções auxiliares

```
1  int comparaCrescente(void *a, void *b)
2  {
3      Base* elemento1 = (Base*)a;
4      Base* elemento2 = (Base*)b;
5      int result;
6
7      if (elemento2->valoresMedida < elemento1->valoresMedida)
8      {
9          return 1;
10     }
11     else if (elemento2->valoresMedida > elemento1->valoresMedida)
12     {
13         return -1;
14     }
15     else
16     {
17         result = strcmp(elemento1->drs, elemento2->drs);
18         if (result == 1) /* elemento2->drs < elemento1->drs */
19         {
20             return 1;
21         }
22         else if (result == 0) /* elemento2->drs > elemento1->drs */
23         {
24             return -1;
25         }
26         else
27         {
```

```
27         {
28             result = strcmp(elemento2->municipio, elemento1->municipio);
29             if (result == 0) /* elemento2->municipio > elemento1->municipio */
30             {
31                 return 1;
32             }
33             else if (result == 1) /* elemento2->municipio < elemento1->municipio */
34             {
35                 return -1;
36             }
37             else
38             {
39                 return 0;
40             }
41         }
42     }
43 }
```


Algoritmo QuickSort Externo

```
1 void quickSortExterno(FILE *arq, int inicio, int fim, int tamArea, int nBytes, int (comp)(void*,void*))
2 {
3     int i = inicio - 1;
4     int j = fim + 1;
5     /* Se a area tiver mais que 1 registro */
6     if(fim - inicio > 1)
7     {
8         /* chama particao para a area do arquivo */
9         particao(arq, inicio, fim, &i, &j, tamArea, nBytes, comp);
10        /* Verifica qual das areas nao ordenadas eh menor */
11        if((i - inicio) > (fim - j))
12        {
13            quickSortExterno(arq, j, fim, tamArea, nBytes, comp);
14            quickSortExterno(arq, inicio, i, tamArea, nBytes, comp);
15        }
16        else
17        {
18            quickSortExterno(arq, inicio, i, tamArea, nBytes, comp);
19            quickSortExterno(arq, j, fim, tamArea, nBytes, comp);
20        }
21    }
22 }
23 }
```

Produto final

C:\Users\quint\Desktop\QuickSort-Externo-TDE\bin\Debug\QuickSort-Externo-TDE.exe

| DRS | Município | Unidade de Saude | Nome de Medida | Valores de Medida |
|------------|--------------------|--|------------------------------|-------------------|
| Araraquara | ARAQUARA | HOSPITAL MUNICIPAL DA MULHER DR JOAO LUIS J ROSSETO | Leitos Ocupados / UTI | 0 |
| Araraquara | ARAQUARA | HOSPITAL MUNICIPAL DA MULHER DR JOAO LUIS J ROSSETO | Leitos Ocupados / Enfermaria | 2 |
| Araraquara | AMÉRICO BRASILENSE | UNIDADE HOSPITALAR DR. JOSÉ NIGRO NETO | Leitos Ocupados / UTI | 0 |
| Araraquara | AMÉRICO BRASILENSE | HOSPITAL ESTADUAL AMÉRICO BRASILENSE | Leitos Ocupados / UTI | 2 |
| Araraquara | AMÉRICO BRASILENSE | UNIDADE HOSPITALAR DR. JOSÉ NIGRO NETO | Leitos Ocupados / Enfermaria | 1 |
| Araraquara | AMÉRICO BRASILENSE | HOSPITAL ESTADUAL AMÉRICO BRASILENSE | Leitos Ocupados / Enfermaria | 1 |
| Araraquara | ARARAQUARA | UPA DR ANTONIO ALONSO MARTINEZ VILA XAVIER | Leitos Ocupados / UTI | 0 |
| Araraquara | ARARAQUARA | UNIDADE DE RETAGUARDA DE URGENCIA E DIAGNOSTICO DO MELHADO | Leitos Ocupados / UTI | 0 |

| | |
|------------------------------|---|
| Leitos Ocupados / Enfermaria | 5 |
| Leitos Ocupados / Enfermaria | 1 |
| Leitos Ocupados / UTI | 0 |
| Leitos Ocupados / Enfermaria | 2 |
| Leitos Ocupados / UTI | 0 |
| Leitos Ocupados / Enfermaria | 3 |
| Leitos Ocupados / UTI | 1 |
| Leitos Ocupados / Enfermaria | 0 |
| Leitos Ocupados / UTI | 0 |
| Leitos Ocupados / Enfermaria | 3 |
| Leitos Ocupados / UTI | 3 |
| Leitos Ocupados / UTI | 3 |
| Leitos Ocupados / UTI | 0 |
| Leitos Ocupados / UTI | 1 |
| Leitos Ocupados / Enfermaria | 1 |
| Leitos Ocupados / Enfermaria | 3 |
| Leitos Ocupados / Enfermaria | 1 |

HU-UFSCAR
HU-UFSCAR

Escolha uma opcao

1. Ordenacao Decrescente

2. Ordenacao Crescente

0. Sair

Agradecemos pela atenção!

Edgar Rocha Abreu

Gabriel de Carvalho Barros

Gustavo Marcos Xavier

Marcio Paiva Barone Borges

Matheus Soléo