

MC102

Algoritmos e Programação de Computadores

Aula de Laboratório 12
Instituto de Computação
Primeiro Semestre de 2012

28 de maio de 2012



Conteúdo

- 1 Problema da Busca
- 2 Manipulação de Arquivos



Busca

Definição

Dada uma coleção de elementos, onde cada elemento possui um identificador, e uma chave para busca. Devemos encontrar o elemento da coleção que possui a mesma chave ou identificar que não existe nenhum elemento com a chave dada.

- Vamos aprender dois algoritmos de busca:
 - **Busca Sequencial;**
 - **Busca Binária;**



Busca Sequencial

- Busca sequencial é bem simples e funciona da seguinte forma:
 - 1 Percorra todo o vetor comparando a **chave** com cada valor do vetor;
 - 2 Se for igual para alguma posição, então devolva esta posição;
 - 3 Caso todas as posições do vetor seja percorrida sem sucesso, devolve -1.



Problema

Problema - Busca sequencial

Faça a função *int buscaSequencial(int vet[], int tam, int chave)* que realize a busca sequencial:

Main()

```
1 #include <stdio.h>
2 int main(){
3     // -1 indica posicao nao usada
4     int pos, vet[]={20,5,15,24,67,45,1,76,-1,-1};
5     pos = buscaSequencial(vet, 8, 45);
6     if(pos != -1)
7         printf("A posicao da chave 45 e:%d\n", pos);
8     else
9         printf("A chave 45 nao esta no vetor!\n");
10 }
```



Problema

Função() - Busca Sequencial

```
1 int buscaSequencial(int vet[], int tam, int chave){  
2     int i;  
3     for(i=0; i<tam; i++){  
4         if(vet[i] == chave)  
5             return i;  
6     }  
7     return -1;  
8 }
```



Busca Binária

- é mais eficiente, mas requer que o vetor esteja ordenado;
- A ideia do vetor é a seguinte:
 - 1 Verifique se a chave de busca é igual ao valor da posição do meio do vetor;
 - 2 Caso seja, devolva esta posição;
 - 3 Caso o valor desta posição seja maior, então repita o processo mas agora considere que o vetor tem metade do tamanho, indo do início até a posição anterior a do meio.
 - 4 Caso o valor desta posição seja menor, então repita o processo mas agora considere que o vetor tem metade do tamanho e inicia na posição seguinte a do meio.



Busca Binária

Exemplo

chave = 5

<i>0</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>
1	2	3	5	6	15

Busca Binária

Exemplo

chave = 5

0	1	2	3	4	5
1	2	3	5	6	15



pIni = 0

pFim = 5

pMeio = $(0+5)/2 = 2$

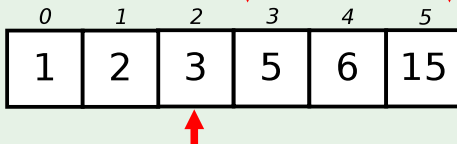


Busca Binária

Exemplo

chave = 5

0	1	2	3	4	5
1	2	3	5	6	15



pIni = 0

pFim = 5

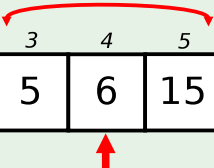
pMeio = $(0+5)/2 = 2$



Busca Binária

Exemplo

chave = 5



0	1	2	3	4	5
1	2	3	5	6	15

pIni = 3

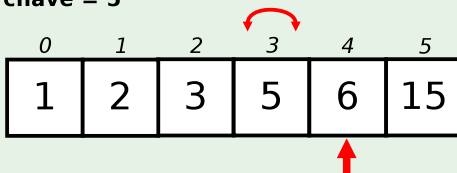
pFim = 5

pMeio = $(3+5)/2 = 4$

Busca Binária

Exemplo

chave = 5



0	1	2	3	4	5
1	2	3	5	6	15

pIni = 3

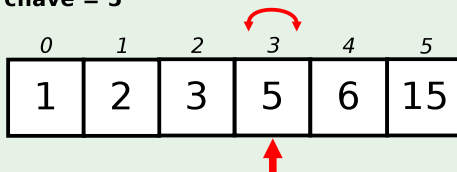
pFim = 5

pMeio = $(3+5)/2 = 4$

Busca Binária

Exemplo

chave = 5



pIni = 3

pFim = 3

pMeio = $(3+3)/2 = 3$

Problema

Function() - Busca binária

```
1  int buscaBinaria(int vet[], int tam, int chave){
2      int posIni=0, posFim=tam-1, posMeio;
3      //enquanto o vetor tiver pelo menos 1 elemento
4      while(posIni <= posFim){
5          posMeio = (posIni + posFim)/2;
6          if(vet[posMeio] == chave)
7              return posMeio;
8          else{    if(vet[posMeio] > chave)
9                  posFim = posMeio - 1;
10             else
11                 posIni = posMeio + 1;
12         }
13     }
14     return -1;
15 }
```



Problema

Main() - Busca binária

```
1  int main(){
2      int vet[] = {1,5,15,20,24,45,67,76,78,100};
3      int pos, i;
4      printf("Vetor Ordenado:");
5      for(i =0; i<10; i++){
6          printf("%d, ", vet[i]);
7      }
8      printf("\n");
9      pos = buscaBinaria(vet, 10, 15);
10     if(pos != -1)
11         printf("A posicao da chave 15 e:%d\n",pos);
12     else
13         printf("A chave 15 na~o esta no vetor!\n");
14 }
```



Arquivos

- Ao utilizar variáveis para guardar valores do programa, essas são armazenadas na **memória (RAM)** e apenas durante a execução do programa;
- E se quisermos armazenar algum dado no **HD**, memória secundária?
- Uma forma é a utilização de arquivos.



Arquivos

- Para se trabalhar com Arquivos em C devemos criar um ponteiro para arquivos:

Ex.: FILE *nome-variavel;

- Após a criação do arquivo podemos associá-lo com um arquivo real do computador usando a função **fopen**:

Ex.:

FILE *arq;

arq = fopen("teste.txt", "r");



Lendo dados do Arquivo

- Para ler dados podemos utilizar a função **fscanf**:
Ex.: `int fscanf(ponteiro, string de formato, variáveis);`

Lendo dados do arquivo.txt

```
char aux;  
FILE *arq = fopen("arquivo.txt", "r");  
fscanf(arq, "%c", &aux);  
printf("%c", aux);
```



Arquivos

Lendo do Arquivo

```
1 #include <stdio.h>
2 int main() {
3     FILE *arq;
4     char aux, nomeArq[100];
5     printf("Entre com nome do arquivo:");
6     scanf("%s", nomeArq);
7     arq = fopen(nomeArq, "r");
8     if (arq == NULL)
9         printf("Erro ao abrir o arquivo: teste.txt");
10    else{
11        printf("----- Dados do arquivo:\n\n");
12        while(fscanf(arq, "%c", &aux) != EOF){
13            printf("%c", aux);
14        }
15    }
16    fclose(arq);
17 }
```

Escrevendo dados do Arquivo

- Para escrevet dados podemos utilizar a função **fprintf**:
Ex.: `int fprintf(ponteiro, string de formato, variáveis);`

Lendo dados do arquivo.txt

```
FILE *fr = fopen("arquivo.txt", "r");  
FILE *fw = fopen("arquivo.txt", "w");  
while (fscanf(fr, "%c", &c) != EOF)  
    fprintf(fw, "%c", c);  
fclose(fr);  
fclose(fw);
```



Arquivos

Escrevendo em Arquivo

```
1 #include <stdio.h>
2 int main() {
3     FILE *arqIn, *arqOut;
4     char aux, nomeArqIn[100], nomeArqOut[100];
5     printf("Entre com nome do arquivo de entrada:");
6     scanf("%s", nomeArqIn);
7     arqIn = fopen(nomeArqIn, "r");
8     if (arqIn == NULL){
9         printf("Erro ao abrir o arquivo: %s\n", nomeArqIn);
10        return 0;
11    }
12    printf("Entre com nome do arquivo de saída:");
13    scanf("%s", nomeArqOut);
14    arqOut = fopen(nomeArqOut, "w");
15    if (arqOut == NULL){
16        printf("Erro ao abrir o arquivo: %s\n", nomeArqOut);
17        return 0;
18    }
19    while(fscanf(arqIn, "%c", &aux) != EOF)
20        fprintf(arqOut, "%c", aux);
21    fclose(arqIn);
22    fclose(arqOut);
23 }
```



Questões?

Obrigado!

Para informação:

Página dos Laboratórios (Tarefas): <http://susy.ic.unicamp.br:9999/mc102ab>

Página do Curso: <http://www.lrc.ic.unicamp.br/~geraldoms/mc102>

E-mail:

geraldoms[at]lrc.ic.unicamp.br

brhenrique.fischer[at]gmail.com

