

Linguagem de Programação II

Prof. Mario Bessa

Aula 11

<http://mariobessa.info>

Busca em vetores

- **Método de Pesquisa:** SEQUENCIAL
- Na pesquisa sequencial ou linear a ideia básica é localizar o elemento procurado através de comparações sucessivas e sequenciais, a partir do primeiro elemento do vetor.
- A pesquisa termina quando o elemento é encontrado ou quando é atingido o fim do vetor.
- No melhor caso, acha-se o elemento procurado na 1ª comparação, no pior caso Na comparação.

Busca Linear

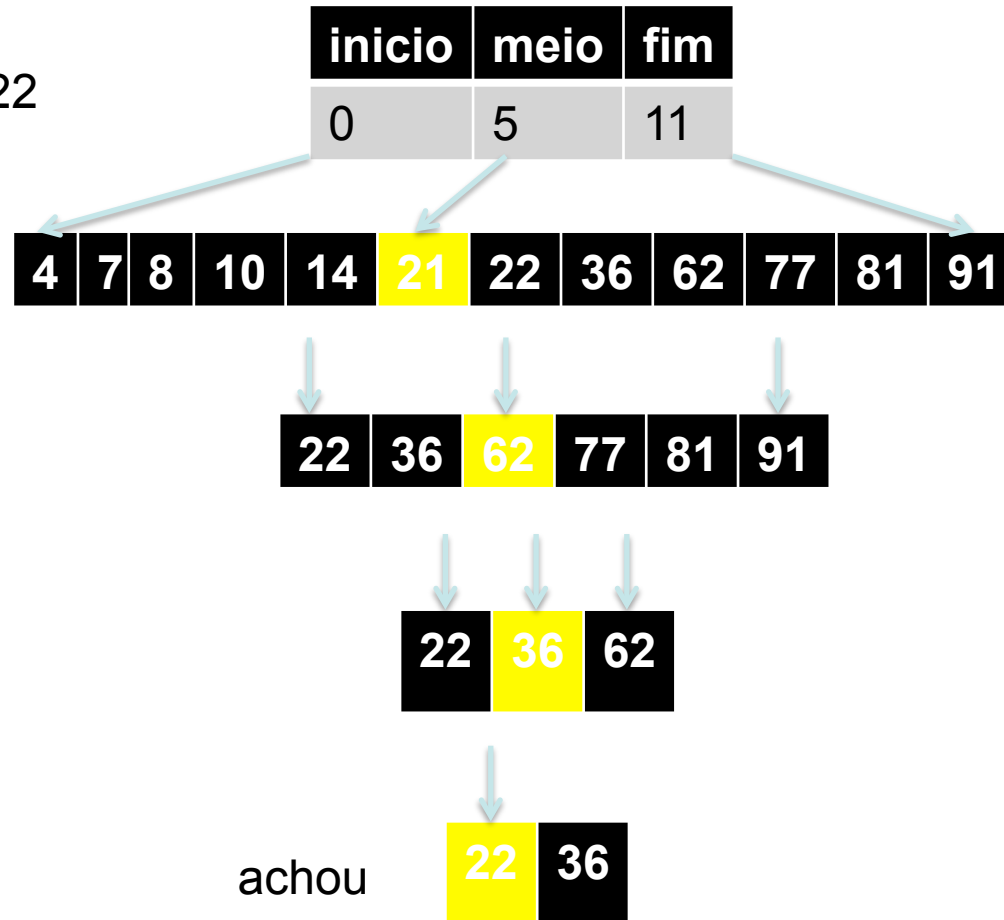
```
#define LIM 4
int main() {
    int    RA[] = {63415, 72282, 74023, 76942};
    float  notas[] = {10.0, 5.5, 7.5, 3.0};
    int i, achou=0, aluno;

    printf("Entre com o RA: ");
    scanf("%d", &aluno);
    for(i=0; i<LIM; i++)
        if(RA[i]==aluno) {
            achou = 1;
            break;
        }
    if(achou)
        printf("Nota: %.2f\n", notas[i]);
    else
        printf("Aluno nao cadastrado\n");
    return 0;
}
```

Busca em vetores

- **Método de Pesquisa:** BINÁRIA
- A ideia básica consiste em diminuir cada vez mais o intervalo de busca. O vetor deve estar ordenado.
- Encontra-se, inicialmente, o elemento central do vetor dividindo-o em dois intervalos (metade).
- O intervalo de pesquisa é o intervalo em que se encontra o elemento. O outro é abandonado. O processo é repetido.
- O término do processo se dá quando o elemento procurado é localizado ou quando o intervalo de busca torna-se vazio.

Elemento=22



Busca Binária

```
#define LIM 4
int main() {
    int    RA[] = {63415, 72282, 74023, 76942};
    float  notas[] = {10.0, 5.5, 7.5, 3.0};
    int i, achou=0, aluno, inicio=0, fim=LIM-1;
    printf("Entre com o código de aluno\n");
    scanf("%d",&aluno); //Lendo ra do aluno.
    while(inicio<=fim) {
        i = (inicio+fim)/2;
        if(RA[i]==aluno) {
            achou = 1;
            break;
        }
        else if(RA[i]>aluno) fim = i-1;
        else                inicio = i+1;
    }
    if(achou) printf("Nota: %.2f\n",notas[i]);
    else      printf("Aluno não cadastrado\n");
    return 0;
}
```

Ordenação por permutação

- **Bubble sort:** trocas são feitas do início para o final.

```
#include <stdio.h>
#define LIM 500
int main(){
    int V[LIM],tmp,troca=0;
    int i,j,n;
    printf("Entre com a quantidade de elementos do vetor ");
    scanf("%d",&n);
    for(i=0; i<n; i++){
        printf("Entre com o elemento %i do vetor ",i);
        scanf("%i",&V[i]);
    }
    for(j=n-1; j>0; j--){
        for(i=0; i<j; i++){
            if(V[i]>V[i+1]){
                troca++;
                tmp = V[i];
                V[i] = V[i+1];
                V[i+1] = tmp;
            }
        }
    }
    printf("\nVetor ordenado usou %i trocas\n",troca);
    for(i=0; i<n; i++)
        printf("%2i",V[i]);
    return 0;
}
```

Vetores

- **Problema:** Faça um programa que leia dois vetores ordenados (*ordem crescente*), com **n** elementos cada, e gere um terceiro vetor com todos elementos mantendo a ordem por intercalação de valores dos outros dois.

X	1	3	5	7	9
---	---	---	---	---	---

Y	2	4	6	8	10
---	---	---	---	---	----

Z	1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	---	----


```

#include <stdio.h>
#define LIM 3
int main(){
    int x[LIM],y[LIM],z[2*LIM];
    int ix=0,iy=0,iz=0;
    printf ("Digite o primeiro vetor\n");
    for (int i=0;i<LIM; i++){
        scanf("%i",&x[i]);
    }
    printf ("Digite o segndo vetor\n");
    for (int i=0;i<LIM; i++){
        scanf("%i",&y[i]);
    }
    while (iz<2*LIM) {
        if (ix<LIM){
            if (x[ix]<y[iy]) {
                z[iz]=x[ix];
                ix++;
                iz++;}
            else if (iy<LIM){
                if (x[ix]>y[iy]) {
                    z[iz]=y[iy];
                    iy++;
                    iz++;

```

```

                }
            }
        }
        else{
            z[iz]=x[ix];
            ix++;
            iz++;
        }
    }
    }
    printf ("Terceiro vetor\n");
    for (int i=0;i<2*LIM; i++)
        printf("%2i",z[i]);
    return 0;
}

```

Vetores

- **Problema:** Faça um programa para preencher dois vetores X e Y (com valores lidos do teclado) de 40 posições. O programa deve atribuir a um vetor Z os valores de X e Y intercalados, de forma que a 1ª posição de Z terá um valor de X (primeira posição), a 2ª posição de Z terá um valor de Y (última posição), a 3ª posição de Z terá um valor de X (2ª posição), a 4ª posição de Z terá o penúltimo elemento de Y e assim sucessivamente até preencher o vetor Z.

X	1	3	5	7	10
---	---	---	---	---	----

Y	12	8	6	4	2
---	----	---	---	---	---

Z	1	2	3	4	5	6	7	8	10	12
---	---	---	---	---	---	---	---	---	----	----

```

#include <stdio.h>
#define LIM 3
int main(){
    int x[LIM],y[LIM],z[2*LIM];
    int ix=0,iy=LIM-1;
    printf ("\nDigite o primeiro vetor ");
    for (int i=0;i<LIM; i++){
        scanf("%i",&x[i]);
    }
    printf ("\nDigite o segundo vetor ");
    for (int i=0;i<LIM; i++){
        scanf("%i",&y[i]);
    }
    for (int i=0; i<2*LIM; i+=2) {
        z[i]=x[ix];
        z[i+1]=y[iy];
        ix++;
        iy--;
    }
    printf ("\nTerceiro vetor ");
    for (int i=0;i<2*LIM; i++){
        printf("%2i",z[i]);
    }
    return 0;
}

```