MC-102 — Aula 16
Busca binária e sequencial

Instituto de Computação – Unicamp

Segundo Semestre de 2011

Roteiro

Vetores — Definição

Coleção de variáveis do mesmo tipo referenciada por um nome comum.

(Herbert Schildt)

- acesso por meio de índice
- posições contíguas na memória
- tamanho pré-definido
- índices fora dos limites podem causar comportamento anômalo do código

Busca - Definição do problema

Problemas

- Dada uma coleção de n elementos, pretende-se saber se um determinado elemento (valor) está presente nessa coleção.
- Determinar a posição desse elemento (valor) em um conjunto de elementos.

Busca - Definição do problema

v= (3, 5, 7, 11, 21, 35, 42, 59, 72)

Problemas

- Como encontrar o valor 82
- Como determinar a posição do valor 21

Busca - Tipos

- Busca sequencial ou linear
- Busca binária

Busca Sequencial

Sequencial

- Percorrer o vetor desde a primeira posição até a última. Para cada posição i, comparamos vetor[i] com valor.
  - Se forem iguais dizemos que valor existe.
  - Se chegarmos ao fim do vetor sem sucesso dizemos que valor não existe.

Busca Sequencial

1. Inicialização

```
i = 0;
encontrado = 0; //falso
```

2. Busca

```
while (i < TAMANHO && !encontrado) {
  if (vetor[i] == valor) {
    encontrado = 1; /*Verdadeiro*/
  } else {
    i++;
  }
}
```

Busca Sequencial

3 Tratamento do Resultado

```
if (encontrado) {
   printf ("Valor %d está na posicao %d\n",
   vetor[i], i);
} else {
   printf ("Valor %d não encontrado\n", valor);
}
```

Veja o programa completo em sequencial.c.

Busca Sequencial - Eficiência

Quanto tempo a busca sequencial demora para executar? Em outras palavras, quantas vezes a comparação vetor[i] == valor é executada?

- Caso valor não esteja presente no vetor, n vezes.
- Caso valor esteja presente no vetor:
  - 1 vez no melhor caso (valor está na primeira posião).
  - n vezes no pior caso (valor está na ultima posição).
  - n/2 vezes no caso médio.

Veja exemplos em seq-random.c e seq-random2.c.

Busca Sequencial

- Implemente uma versão recursiva para este algoritmo.
- Qual é o caso base? E o recursivo?

Veja o exemplo em sequencial-rec.c.

Busca Binária

- Agora vamos supor que o vetor está ordenado (crescente).
- Sera que é possível resolver o problema de modo mais eficiente?
- Será que podemos fazer algo melhor do que verificar cada posição do vetor?
- Que tal verificarmos se a posição do meio do vetor contem x. O que podemos concluir?

Busca Binária

- Se x for menor do que o elemento do meio, então x deve estar na metade inferior do vetor.
- Se x for maior do que o elemento do meio, então x deve estar na metade superior do vetor.

Em outras palavras

- Caso a lista esteja ordenada, sabemos que, para qualquer i e j, i $<$ j, se, e somente se, A[i] $<=$ A[j].
- Portanto, comparando um determinado elemento com o elemento procurado, saberemos:
  - Se o elemento procurado é o elemento comparado;
  - Se ele está antes do elemento comparado ou depois;

Busca Binária

Solução: busca binária

- Se fizermos isso sempre com o elemento do meio da lista, a cada comparação dividiremos a lista em duas, reduzindo nosso tempo de busca.
- Se em um determinado momento o vetor, após sucessivas divisões, tiver tamanho zero, então o elemento não está no vetor.

Busca Binária

1 Inicialização

```
int direita, esquerda, meio;
encontrado = 0; /*Falso*/

esquerda = 0;
direita = TAMANHO - 1;
```

Busca Binária

2 Busca

```
while(esquerda<=direita && !encontrado){
        meio=(direita+esquerda)/2;
        if (vetor[meio] == valor)
              encontrado = 1; /*Verdadeiro*/
        else if (valor < vetor[meio])
              direita = meio - 1;
        else
              esquerda = meio + 1;
}
```

Busca Sequencial

3 Tratamento do Resultado

```
if(encontrado){
        printf ("Valor %d encontrado na posicao %d\n",
        vetor[meio], meio);
}
else{
        printf ("Valor %d não encontrado\n", valor);
}
```

Veja o programa completo em binaria.c.

Busca Binária - Eficiência

Quanto tempo a busca binária demora para executar? Em outras palavras, quantas vezes a comparação vetor[i] == valor é executada?

- Caso valor não exista no vetor, log2 (n) vezes.
- Caso valor exista no vetor:
  - 1 vez no melhor caso (valor é a mediana do vetor).
  - log2 (n) vezes no caso m ?dio.
- O logaritmo de base 2 aparece porque sempre dividimos o intervalo ao meio.

Veja um exemplo em bin-random.c.

Comparação entre Busca sequencial e binária

- Para n = 1000, o algoritmo de busca sequencial irá executar 1000 comparações no pior caso, 500 operações no caso médio.
- Por sua vez, o algoritmo de busca binária irá executar 10 comparações no pior caso, para o mesmo n.
- O algoritmo de busca binária supõe que o vetor está ordenado. Ordenar um vetor também tem um custo, superior ao custo da busca sequencial.

Busca Binário

- Implemente uma versão recursiva para este algoritmo.
- Qual é o caso base? E o recursivo? Veja o exemplo em binario-rec.c.

Busca Binário

- Faça uma função que realize uma busca binária
- Faça uma função que realize uma busca sequencial