Laboratório de Algoritmos e Técnicas de Programação II

Aula 06 - Algoritmos de busca: busca sequencial e busca binária

Álvaro Magri Nogueira da Cruz

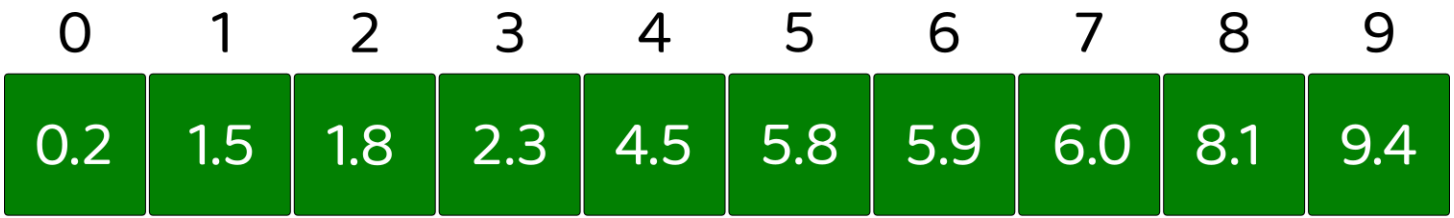
Agenda

Introdução I

Observações

- Com grandes quantidades de dados armazenados em *arrays* fica difícil adivinhar a posição de um elemento;
- Talvez seja necessário determinar se um array contém um valor que combina com certo valor de chave;
- O processo de encontrar determinado elemento de um array é chamado **pesquisa**;
- Discutiremos duas técnicas de pesquisa:
  - **Pesquisa linear simples**;
  - **Pesquisa binária**.

Introdução II



Chave de pesquisa: **4.5**

Posição da chave: **4**

Pesquisa sequencial ou linear simples I

A pesquisa

- A pesquisa linear compara cada elemento do *array* com a chave de pesquisa;
- Como o *array* não está em uma ordem em particular, o valor pode ser encontrado:
  - Tanto no primeiro elemento;
  - Quanto no último.
- Na média, portanto, o programa terá de comparar a chave de pesquisa com metade dos elementos do *array*.

Pesquisa sequencial ou linear simples II

```
1 int linearSearch(int array[], int key, int size){
2     int n;
3     for(n=0; n<size; n++){
4         if(array[n]==key){
5             return n;
6         }
7     }
8     return -1;
9 }
```

Pesquisa binária I

Observação

- O método de pesquisa linear funciona bem para *arrays* **pequenos** ou **não ordenados**;
- Entretanto, para *arrays* **grandes**, a pesquisa linear é **ineficaz**;
- Se o *array* estiver **ordenado**, a técnica de pesquisa binária de alta velocidade poderá ser utilizada.

Pesquisa binária II

A técnica

- O algoritmo de pesquisa binária **desconsidera metade dos elementos** em um *array* ordenado após cada comparação.
- O algoritmo localiza o elemento **do meio** do *array* e o compara com a chave de pesquisa;
  - Se são iguais, foi encontrada, e a posição do elemento é retornada;
  - Se não são iguais, o problema reduz à pesquisa da metade do array.
    - Se a chave de pesquisa for menor que o elemento do meio do array, a primeira metade do array é pesquisada;
    - Caso contrário, a segunda metade do array é pesquisada.
    - Se a chave de pesquisa não é encontrada no subarray especificado (parte do array original), o algoritmo é repetido em um quarto do array original;
    - E assim, prossegue até chegar a um único elemento.

Pesquisa binária III

Pesquisa binária IV

```
1  int binarySearch(int array[], int key, int low, int high){
2      int middle;
3      while (low <= high) {
4          middle = (low + high)/2;
5          if (key == array[middle]){
6              return middle;
7          }
8          else if (key < array[middle]){
9              high = middle - 1;
10         }
11         else {
12             low = middle + 1;
13         }
14     }
15     return -1;
16 }
```

Exercícios I

1. Crie um programa que receba um vetor INTEIRO ORDENADO (pelo menos 10000 posições, aconselho usar preenchimento automático). Este vetor deverá ser armazenado em um arquivo para posterior consulta. Leia o arquivo para recuperar o vetor e permita que o usuário faça uma busca por um número. Deixe que o usuário escolha o método de busca (sequencial ou binário). Colha o tempo de busca e avise o usuário o quanto ele gastaria de tempo se tivesse optado pela outra estratégia.
2. **DESAFIO:** o método de busca binária é CLARAMENTE plausível de recursão. Como desafio, crie uma função recursiva para busca binária.

Exercícios II

3. Crie um programa que receba um vetor INTEIRO ORDENADO (pelo menos 10000 posições). Permita ao usuário buscar um número. Faça a busca utilizando os três métodos: sequencial, binário e binário recursivo. Faça o comparativo entre os tempos de execução.

Referências I

1. **Deitel H. M., e Deitel P.J; "C: Como programar". 6.ed. Pearson Prentice Hall, 2011. 818p.**
2. Jean Paul Tremblay & Richard P. Bunt. "Ciência dos Computadores - Uma abordagem algorítmica". McGraw-Hill.
3. Jaime Evaristo. "Aprendendo a Programar / Programando em Turbo Pascal". Edufal - Editora da Univ. Federal de Alagoas. Maceió, 1996.
4. Harry Farrer et al. "Pascal Estruturado (da série "Programação Estruturada de Computadores")". Editora Guanabara Dois. Rio de Janeiro, 1985.
5. Stephen O'Brien. "Turbo Pascal 6 Completo e Total". Makron Books.

Referências II

6. Celes, W., Cerqueira, R., Rangel, J.L. "Introdução a Estrutura de Dados". Elsevier, 2004.
7. Feofiloff, P. "Algoritmos em Linguagem C". Elsevier, 2009. 208p.
8. Schildt, H. "C Completo e Total". 3ª ed. Pearson. 1996. 852p.