Problem Solving With C - UE24CS151B

Searching using C

Prof. Sindhu R Pai
PSWC Theory Anchor, Feb-May, 2025
Department of Computer Science and Engineering

PROBLEM SOLVING WITH C

Searching

1. Points for Discussion
2. Introduction
3. Searching Algorithms
4. Linear Search
5. Binary Search - Pictorial Representation and Implementation

PES UNIVERSITY

Points for Discussion!

- Have you spent a day without searching for something?
- Finding a particular item among many hundreds, thousands, millions or more.
  - Scenario: Finding someone's phone number in our phone
- What if one wants to save the time consumed in looking to each item in a collection of items?

Introduction

- Identifying or finding a particular record/item/element in a collection of records/items/elements and knowing the place of it
- Collection/Group where searching must be done may be sorted or unsorted
- Search may be Successful search or Unsuccessful based on the availability of the record/item/element in a collection

Searching Algorithms

- Random Search
- Sequential or Linear search
- Non - Sequential or Binary Search.

Linear Search

- Performs search on any kind of data
- Starts from 0^{th} item till the end of the collection

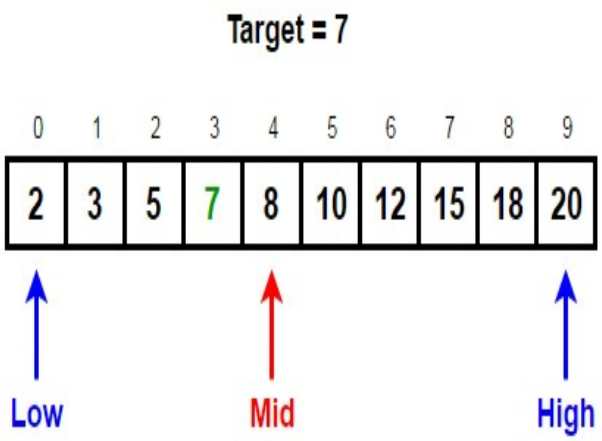
One-Dimensional Array having 7 Elements

- Coding Example

Binary Search

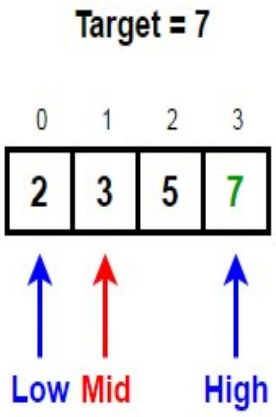
- Necessary condition: **Collection of data should be sorted**
- **Begins comparison at the middle** of the collection
  - If matched, return the index of the middle element
  - If not matched, check whether the element to be searched is lesser or greater than the middle element
- If the element to be searched is greater than the middle element, pick the elements on the right side of the middle element and repeat from the start
- If the element to be searched is lesser than the middle element, pick the elements on the left side of the middle element and repeat from the start

Pictorial Representation of Binary Search



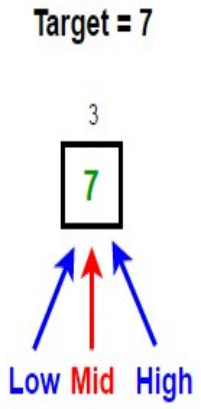
Since 8 (Mid) > 7 (target),
we discard the right half and go **LEFT**

New High = Mid - 1



Since 3 (Mid) < 7 (target),
We discard the left half and go **RIGHT**

New low = mid + 1



Now our search space consists of only one element 7. Since 7 (Mid) = 7 (target), we return index of 7 i.e. 3 and terminate our search

Binary Search Implementation

Given a sorted Array A of n elements and the target value is T

Iterative Algorithm

1. Set L: 0 and R: n-1
2. If(L>R), Unsuccessful Search
3. Else Set m: (L+R)/2 // m: position of middle element
4. If A_m < T, set L to m+1 and go to step 2
5. If A_m > T, set R to m-1 and go to step 2
6. If A_m is T, search done, return m

Recursive Algorithm

BinarySearch(T, A)

1. Set L: 0 and R: n-1
2. If(L>R), return -1 // Unsuccessful Search
3. Else Set m: (L+R)/2 // m: position of middle element
4. If A_m is T, search done, return m
5. If A_m < T, return BinarySearch(T, A_0 to A_{m-1})
6. Else return BinarySearch(T, A_{m+1} to A_{n-1})

THANK YOU

Department of Computer Science and Engineering

Dr. Shylaja S S, Director, CCBD & CDSAML, PESU
Prof. Sindhu R Pai - sindhurpai@pes.edu

Ack: Teaching Assistant - U Shivakumar