SEARCHING ALGORITHMS

-PINAK PATEL

Searching in Linear Array

- The process of finding a particular element of an array is called "Searching".
- If the item is not present in the array, then the search is unsuccessful.
- There are two types of search,
  - Linear search
  - Binary Search

Linear Search

- The linear search compares each element of the array with the search key until the search key is found.
- To determine that a value is not in the array, the program must compare the search key to every element in the array.
- It is also called "Sequential Search" because it traverses the data sequentially to locate the element.

Linear Search

- **Algorithm: (Linear Search)**
- LINEAR (A, SKEY)
- Here **A is a Linear Array with N elements and SKEY is a given item** of information to search. This algorithm finds the location of SKEY in **A and if successful, it returns its location otherwise it returns -1 for** unsuccessful.

1. Repeat for i = 0 to N-1

2. if(A[i] = SKEY) return i [Successful Search]

[End of loop]

3. return -1 [Un-Successful]

4. Exit.

Binary Search

- It is useful for the large sorted arrays.
- The binary search algorithm can only be used with ***sorted array and eliminates one half of the elements in the array*** being searched after each comparison.
- The algorithm locates the middle element of the array and compares it to the search key.
- If they are equal, the search key is found and array subscript of that element is returned.
- Otherwise the problem is reduced to searching one half of the array.
- If the search key is less than the middle element of array, the first half of the array is searched.
- If the search key is not the middle element of in the specified sub array, the algorithm is repeated on one quarter of the original array.
- The search continues until the sub array consist of one element that is equal to the search key (search successful).
- But if Search-key not found in the array then the value of END of new selected range will be less than the START of new selected range.

Binary Search Example

Search Key=22	
A[0]	3
A[1]	5
A[2]	9
A[3]	11
A[4]	15
A[5]	17
A[6]	22
A[7]	25
A[8]	32
A[9]	54

Start=0

End = 9

Mid=int(Start+End)/2

Mid= int (0+9)/2

Mid=4

Start=4+1 = 5

End = 9

Mid=int(5+9)/2 = 7

Start = 5

End = 7 – 1 = 6

Mid = int(5+6)/2 =5

Start = 5+1 = 6

End = 6

Mid = int(6 + 6)/2 = 6

Found at location 6

Successful Search

Binary Search Example

Search Key=8	
A[0]	3
A[1]	5
A[2]	9
A[3]	11
A[4]	15
A[5]	17
A[6]	22
A[7]	25
A[8]	32
A[9]	54

Start=0

End = 9

Mid=int(Start+End)/2

Mid= int (0+9)/2

Mid=4

Start=0

End = 3

Mid=int(0+3)/2 = 1

Start = 1+1 = 2

End = 3

Mid = int(2+3)/2 =2 5

Start = 2

End = 2 – 1 = 1

End is < Start

Un-Successful Search

Binary Search Algorithm

- Here **A is a sorted Linear Array with N elements and SKEY is a given item** of information to search. This algorithm finds the location of SKEY in **A and if successful, it returns its location otherwise it returns -1 for** unsuccessful.
- Binary Search (A, SKEY)

1. [Initialize segment variables.]

- Set START=0, END=N-1 and MID=INT((START+END)/2).

2. Repeat Steps 3 and 4 while START ≤ END and A[MID]≠SKEY.

3. If SKEY< A[MID]. then

- Set END=MID-1.

Else

- Set START=MID+1.

[End of If Structure.]

4. Set MID=INT((START +END)/2).

[End of Step 2 loop.]

5. If A[MID]= SKEY then

- **Set LOC= MID**

Else:

- Set LOC = -1

[End of IF structure.]

6. return LOC and Exit

Computational Complexity of Binary Search

- The ***Computational Complexity of the Binary Search algorithm is measured*** by the maximum (worst case) number of Comparisons it performs for searching operations.
- The searched array is divided by 2 for each comparison/ iteration.
- Therefore, the maximum number of comparisons is measured by: ***log2(n) where n is the size of the array***

- **Example:**
- If a given sorted array 1024 elements, then the maximum number of comparisons required is:
- *log2(1024) = 10 (only 10 comparisons are enough)*

Computational Complexity of Linear Search

- Note that the ***Computational Complexity of the Linear Search is the*** maximum number of comparisons you need to search the array.
- As you are visiting all the array elements in the worst case, then, the number of comparisons required is:
- ***n (n is the size of the array)***

- **Example:**
- If a given an array of 1024 elements, then the maximum number of comparisons required is:
- ***n-1 = 1023 (As many as 1023 comparisons may be required)***


Thank You !
Questions ?