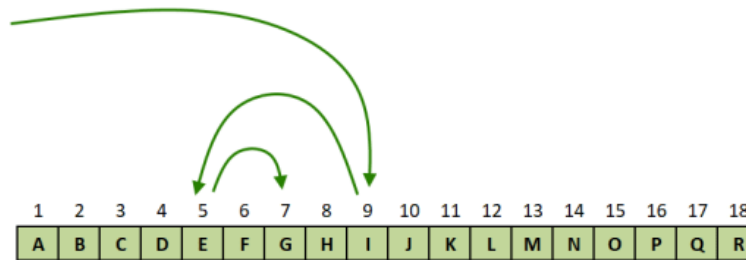
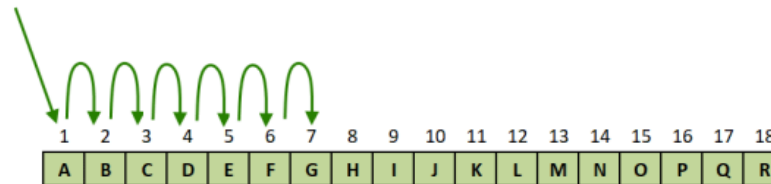


Linear and Binary Search



Binary Search - Find 'G' in sorted list A-R



Linear Search - Find 'G' in sorted list A-R

Searching algorithms

- Given a list, find a specific element in the list
- We will see two types
 - Linear search (sequential search)
 - Binary search

Linear search

- Given a list, find a specific element in the list
 - List does NOT have to be sorted!

procedure linear_search (x : integer; a_1, a_2, \dots, a_n : integers)

$i := 1$

while ($i \leq n$ and $x \neq a_i$)

$i := i + 1$

if $i \leq n$ **then** $location := i$

else $location := 0$

{ $location$ is the subscript of the term that equals x , or it is 0 if x is not found}

Linear search running time

- How long does this take?
- If the list has n elements, worst case scenario is that it takes n “steps”
 - Here, a step is considered a single step through the list

Binary search

- Given a list, find a specific element in the list
 - List MUST be sorted!
- Each time it iterates through, it cuts the list in half

procedure binary_search (x : integer; a_1, a_2, \dots, a_n : increasing integers)

$i := 1$ { i is left endpoint of search interval }

$j := n$ { j is right endpoint of search interval }

while $i < j$

begin

$m := \lfloor (i+j)/2 \rfloor$ { m is the point in the middle }

if $x > a_m$ **then** $i := m+1$

else $j := m$

end

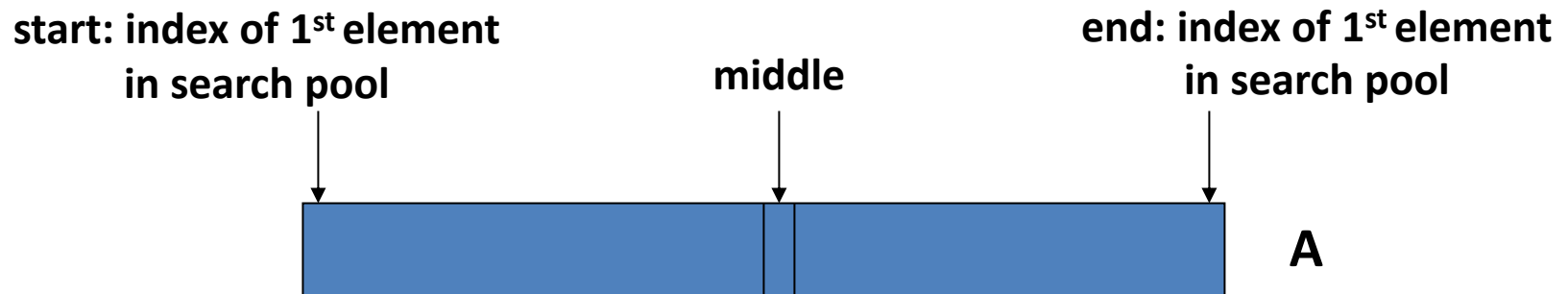
if $x = a_i$ **then** $location := i$

else $location := 0$

{*location* is the subscript of the term that equals x , or it is 0 if x is not found}

Binary Search

- Search a sorted array for a given value



- $BS(A, key, start, end)$
 - Look for *key* in the array *A*
 - where elements are sorted according to ascending order
 - A method that calls itself with a smaller input set

Binary search recursion: pseudo-code

```
boolean BS(int[] A, key, start, end)
    mid = (start+end)/2
    if(A[mid] == key)
        return true
    else
        if(end <= start)
            return false
        else
            if (A[mid] > key)
                return BS(A, key, start, mid-1)
            else
                return BS(A, key, mid+1, end)
```

Binary search running time

- How long does this take (worst case)?
- If the list has 8 elements
 - It takes 3 steps
- If the list has 16 elements
 - It takes 4 steps
- If the list has 64 elements
 - It takes 6 steps
- If the list has n elements
 - It takes $\log_2 n$ steps