Analysis of Algorithms Part 1: Binary Search

Sriram Sankaranarayanan

Binary Search : Problem Description

Problem: Does the given element belong to the sorted list of numbers?

```
Input:
  sorted list of numbers: [1, 6, 7, 19, 22, 25, 31, 55]
  element: 18
Expected Answer:
  False (element is not in the list).

Input:
  sorted list of numbers: [1, 6, 7, 14, 17, 21, 25]
  element: 6
Expected Answer:
  True, Index : 1
```

Binary Search: Basic Idea

Search for element: 6

```
[1, 6, 7, 14, 17, 21, 25]
```

Binary Search: Basic Idea # 2

Search for element: 18

```
[1, 6, 7, 14, 17, 21, 25]
```

Binary Search Implementation

```
def binarySearchHelper( lst, elt, left, right)
    # Requirements:
    # 0 <= left <= right < size(lst)
    # Invariant:
    # If elt is found in lst, it must be found in the
    #    sub-list [ lst[left],…,lst[right] ]
```

Binary Search Implementation

```
def binarySearchHelper( lst, elt, left, right)
   # Requirements:
   # 0 <= left <= right < size(lst)
   # Invariant:
   # If elt is found in lst, it must be found in the
   #    sub-list [ lst[left],…,lst[right] ]
   if (left > right):
        return None # Search region is empty -- let us bail.
    else:
         mid = (left + right)//2 # Note that // is integer division
        if lst[mid] == elt:
            return mid # BINGO -- we found it. Return its index and that we found it
        elif lst[mid] < elt:
            return binarySearchHelper(lst, elt, mid+1, right)
        else: # lst[mid] > elt
            return binarySearchHelper(lst, elt, left, mid-1)
```

Binary Search

```
def binarySearch(lst, elt)
    binarySearchHelper(lst, elt, 0, size(lst) - 1)
```

Correctness of Binary Search

`binarySearchHelper(lst, elt, left, right)` has the following behavior for lst sorted in ascending order:

- If elt belongs to the sub-list `lst[left]…list[right]`, it returns the index of elt in lst.
- If elt does not belong to the sub-list, it returns the special value None.

How to prove correctness?

Running Time Analysis of Binary Search

Running Time Analysis of Binary Search (Cont)