

Lecture 27. Basics Counting Operations in Algorithms II

Example 27.1. Consider the recursive binary search algorithm

```
def binarySearchRecursive(alist, item):  
    if len(alist) == 0:  
        return False  
    else:  
        midpoint = len(alist)//2    # The operator //  
                                     # is equivalent to the floor function.  
        if alist[midpoint] == item:  
            return True  
        else:  
            if item < alist[midpoint]:  
                return binarySearchRecursive(alist[:midpoint], item)  
            else:  
                return binarySearchRecursive(alist[midpoint+1:], item)
```

- (a) Let $X = \{2, 3, 5, 8, 13, 21, 34, 55, 70\}$. Evaluate `binarySearchRecursive(X, 55)`.
- (b) Find a recurrence relation of counting number of comparisons in the algorithm
- (c) Use (b) to approximate the worst-case complexity of the recursive binary search function.