

## CS200 - Recurrence Worksheet

1. Create a recurrence relation for the worst case runtime of the following algorithm for binary search when the array is initially size  $n$ . You may assume  $n$  is a power of 2. Use the iterative method to solve the recurrence relation.

**Algorithm 1: BinarySearch( $A, V$ )**

**Input** : (1) Array (list)  $A$  containing integers, where there are no repeated integers and the integers are sorted from smallest to largest, (2) an element  $V$

**Output:** True if there is an index  $g$  such that  $A[g] = V$  and False otherwise.

```
1  $l$ =length of  $A$ ;  
  // Base Case  
2 if  $l = 1$  and  $A[1] = V$  then  
3 | return True;  
4 else  
5 | return False;  
6 end  
  // Recursive step  
7  $mid = \lfloor l/2 \rfloor$ ;  
8 if  $A[mid] < V$  then  
9 | return BinarySearch( $A[mid + 1 : l], V$ );  
10 else  
11 | return BinarySearch( $A[1 : mid], V$ );  
12 end
```

2. Create a recurrence relation for the number of ways a person can climb  $n$  stairs if the person can take one stair or two stairs at a time. (For this problem, order matters, so if the person takes three steps by taking the first step by itself and the next two together, that is different than if the person takes the first two steps together, and the third by itself. Think about the options for the very last step.) How many ways can this person climb a flight of 5 stairs?
3. Let  $T(n, k)$  be the number of strings in  $\{0, 1, 2\}^n$  whose digits sum to the number  $k$ . Create a recurrence relation for  $T(n, k)$ .