CS200 - Recurrence Worksheet

1. Create a recurrence relation for the worst case runtime of the following algorithm for binary search when $f - s + 1 = n$. You may assume n is a power of 2. Use the iterative method to solve the recurrence relation.

> **Algorithm 1:** BinarySearch(A, x, s, f)
>
> **Input** : Sorted (in increasing order) array of integers A, an integer x that occurs in the array, a starting index s and an ending vertex f
>
> **Output:** An index i such that $A[i] = x$.
>
> 1 **if** $s == f$ **then**
> 2   return s;
> 3 **end**
> 4 $mid = \lfloor (s + f)/2 \rfloor$;
> 5 **if** $A[mid] < x$ **then**
> 6   return BinarySearch$(A, x, mid + 1, f)$
> 7 **else**
> 8   return BinarySearch(A, x, s, mid)
> 9 **end**

2. Let $K(n)$ be the size of the set of n-digit numbers that have an even number of 0's. Create a recurrence relation for $K(n)$. What is $K(3)$? (Hint 0: zero 0's is an even number of 0s. Hint 1: think about the possible options for the value of the final digit of the number if you know the number of options for the first $n-1$ digits. Hint 2: The size of the set of numbers that *don't* have an even number of 0's is the total number of elements minus the set of numbers that *do* have an even number of 0's.)

3. Create a recurrence relation for the number of ways a person can climb n stairs if the person can take one stair or two stair at a time. How many ways can this person climb a flight of 8 stairs? (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.)