

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, 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 index 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
```

Solution Let $T(n)$ be the run time of the algorithm, when n is the size of the initial array. Then

$$T(n) = T(n/2) + C \tag{1}$$

where C is some constant because each time we run the algorithm, we make one recursive call on an input of size $n/2$. Using the definition of T , we have that this call take time $T(n/2)$.

The initial conditions are

$$T(1) = B \tag{2}$$

where B is some constant.

Then using the iterative approach, we have

$$\begin{aligned} T(n) &= T(n/4) + C + C = T(n/4) + 2C \\ &= T(n/8) + C + 2C = T(n/8) + 3C. \end{aligned} \tag{3}$$

We see the pattern is

$$T(n) = T(n/2^k) + k \times C. \quad (4)$$

We are interested in the point where $T(n/2^k)$ becomes $T(1)$, because we know $T(1)$. Now $n/2^k = 1$ when $2^k = n$, which happens when $k = \log_2(n)$. Plugging this value of k into the above expression, we have

$$T(n) = T(1) + \log_2(n) \times C = O(\log_2(n)). \quad (5)$$

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 stair at a time. How many ways can this person climb a flight of 5 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. Think about the options for the very last step.)

Solution Let $T(n)$ be the number of ways the person can take n stairs.

If the person takes n stairs, they could either take the last stair on it's own, or they could take the last two stairs together. If they take the first stair as one step, there are $T(n - 1)$ ways that they could take the first $n - 1$ stairs to get there, so there are $T(n - 1)$ ways of taking the final stair alone. If they take the final two stairs together, there are $T(n - 2)$ ways that they could have gotten to the $(n - 2)$ th stair, and then one way that they can take the final two together.

Thus

$$T(n) = T(n - 1) + T(n - 2). \quad (6)$$

For the base cases, we have

$$T(1) = 1 \quad (7)$$

because there is only one way to take one stair, and

$$T(2) = 2 \quad (8)$$

because the person could either take the 2 stairs at once, or could go up one at a time.

Now for the recurrence relation, So

$$T(5) = T(4) + T(3) = T(3) + T(2) + T(2) + T(1) = T(3) + 5 = T(2) + T(1) + 5 = 8 \quad (9)$$

3. 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.)

Solution The base case is $K(1) = 9$ since the only way you can have a 1-digit number with an even number of 0s is to have no 0s.

For the recurrence relation, We can break up the set of n -digit numbers with an even number of 0s into those that have a 0 in the final position, and those that don't. For those that have a final 0, in order to have an even number of 0s total, there must be an odd number of 0s in the first $n - 1$ digits. The number of $(n - 1)$ -digit numbers with an odd number of 0s is the total number of $(n - 1)$ digit numbers (which is 10^{n-1}), minus the number of $(n - 1)$ -digit numbers with an even number of 0s (which is $K(n - 1)$). Thus the number of n -digit numbers with an even number of 0s and the last digit equal to 0 is

$$10^{n-1} - K(n - 1). \quad (10)$$

In the case that the final digit is not a zero, then for there to be an even number of 0s, we must have an even number of zeros in the first $n - 1$ digits. There are $K(n - 1)$ such numbers. Since we have a choice of 9 digits for the last digit, there are $9 \times K(n - 1)$ numbers that don't have a final digit 0. Thus the number of n -digit numbers with an even number of 0s and the last digit equal to 0 is

$$9K(n - 1). \quad (11)$$

Putting it all together, we have

$$K(n) = 9K(n - 1) + 10^{n-1} - K(n - 1) = 8K(n - 1) + 10^{n-1}. \quad (12)$$

Thus

$$K(3) = 8K(2) + 10^2 = 8(8K(1) + 10^1) + 10^2 = 8 \times 8 \times 9 + 180 = 756. \quad (13)$$