Christopher Singh October 8, 2019

Solving Recurrences

1. $T(n)=2 * T(n/2) + 1000n \quad \forall n \geq 2 \qquad T(1) = 1$

log n n → This function has a Big O that is: **O(n log n)**

Assumption: T(n) ∈ O(n log n)

Step 1: Base case: n = 2

T(2) = 2 * T(2/2) + 1000(2) <= c2log2
2 * 1 + 2000 <= c2log2

2002 <= c2log2
1001 <= c

n = 2 and c = 1001

Step 2: T(k) <= c*k*log k

Step 3: T(n) = 2T(n/2) + 1000n <= c*n*log n let k = n/2

$2*c*\frac{n}{2} * \log(\frac{n}{2}) + 1000n <= c*n* \log n$

c*n (log n – log 2) + 1000n <= c*n* log n
c*n (log n – 1) + 1000n <= c*n* log n

c*n*log(n) – cn + 1000n <= c*n* log n

c*n*log(n) – (cn – 1000n) <= c*n*log n

2. $T(n)=7*T(n/2)+18n^2 \quad \forall n \geq 2 \qquad T(1) = 1$

log n n^2 → This function is: **O($n^2 \log n$)**

Assumption: T(n) ∈ O($n^2 \log n$)

Step 1: Base Case: (n = 2)

$T(2) = 7 * T(2 / 2) + 18(2)^2 <= c2^2 \log 2$
79 <=c4 log 2
19.75 <= c

n = 2 and c = 19.75

Step 2: T(k) = $ck^2 log\ k$

Step 3: T(n) = 7 * T ($\frac{n}{2}$) + $18n^2$ <= $cn^2 log\ n$

7*c ($\frac{n}{2}$)^2 log($\frac{n}{2}$) + $18n^2$<= $cn^2 log\ n$

7 * c * $\frac{n^2}{4}$ * (log(n) – log(2)) + $18n^2$<= $cn^2 log\ n$

1.75cn^2 $*$ $(log\ (n)\ -1\ )$ + $18n^2$<= $cn^2 log\ n$

1.75c$n^2 log(n)$ $-$ 1.75cn^2 + $18n^2$<= $cn^2 log\ n$

1.75c$n^2 log(n)$ - $n^2(1.75c\ +\ 18)$ <= $cn^2 log\ n$

Note: 1.75 is just a constant and does not affect the Big O of a function.

We are left with: $\boldsymbol{cn^2 log(n)}$ **-** $\boldsymbol{n^2(1.75c\ +\ 18)}$ **<=** $\boldsymbol{cn^2 log\ n}$

3.
```
public int recursive(int n){
      int sum=0;
      for(int i=1; i<= n; i++){
            sum= sum +1;
      }
      if(n>1){return recursive(n/2);}
      else{return1}
}
```

The for loop has a runtime of n and the recursive call has a runtime of n/2. The entire code has the Big O of: **O(n log n)**

T(n) = 2T($\frac{n}{2}$) $+$ n $\forall n \geq 2$ where T(1) = 1

Assumption: T(n) $\in$ O(n log n)

Step 1: Base case: n = 2

T(2) = 2T(2/2) + 2 <= c2log 2

4 <= c2

2 <= c

n =2 and c = 2

Step 2: T(k) <= c*k*log k

Step 3: let k = (n/2)

T(n) = $2T(\frac{n}{2}) + n$ <= c*n*log n

$2c\frac{n}{2}\log(\frac{n}{2})$ + n <= c*n*log n

cn(log (n) – log 2) + n <= c*n*log n

cn (log n – 1) + n <= c*n*log n

c*n*log n – cn + n <= c*n*log n

Last Step: c*n*log n – (cn -n) <= c*n*log n