CS2302 Data Structures

Fall 2020

Post-Review Session

Master Theorem

$T(n)=O(n^{\log_b a})$ if $a>b^k$
$T(n)=O(n^k \log n)$ if $a=b^k$
$T(n)=O(n^k)$ if $a<b^k$

1. What is the recurrence equation that describes the running time of the following recursive function? T(n) = a T(n / b) + n ^ k. What are the values of a, b, and k? What is their time complexity?

```
def m0(a):
    if len(a)>0:
        i = 2
        mid = len(a)//2
        while i>0:
            m0(a[mid+1:])
            m0(a[mid+1:])
            print(A[mid])
            m0(a[:mid])
            m0(a[:mid])
            i-=1

def m1(a):
    if len(a)>0:
        for i in range(1,3):
            m1(a[:len(a)//2])
        for i in range(len(a)):
            for j in range(len(a)):
                print(a)
```

2. What is the Time Complexity of the following methods?

```
def m2(a):
    if len(a)>0:
        print(a[0])
        m2(a[1:])
        m2(a[1:])

def m3(a):
    if len(a)>0:
        print(a[-1])
        m2(a[:-1])

def m4(a):
    n = len(a)
    while n>0:
        print('Hey')
        n=n//2
```

```
def m5(a):
    for i in range(10):
        print('Hello, World!')
```

3. What is the Time complexity of the following recurrence equations?

$T(n)=16T(n/2)+n^4$
$T(n)=4T(n/4)+n^2$

4. Write the function odd_cols_and_reverse(I) that receives a numpy array and returns an sub-array of I containing only the even rows and a reverse version of the array. For example:

```
Original array:
 [[ 0  1  2  3  4]
 [ 5  6  7  8  9]
 [10 11 12 13 14]
 [15 16 17 18 19]]
Result:
 [[ 4  3  2  1  0]
 [14 13 12 11 10]]
```

5. Write a **recursive** function *not_multiples(L,k)* that receives a Python list L and an integer k and returns the number of elements in L that are not multiples of k.

6. Write the function *swap_first_and_third(L)* that receives a reference to a List object L and swaps the data inside the first not with the data inside the third node.

7. Given an integer array *nums* and an integer target, determine if it is possible to divide *nums* in two groups, so that the sum of all the values are equal to the target. Note: Feel free to create a helper method.

Extra Question.
Complete the function top_bottom_circles() by creating a call stack and simulating the creation and deletion of activation records. This method will plot the following figures: