**Algorithm Analysis and Complexity Theory Assignment 2 Solutions**

**Tolga Gölbaşı**

**1-**T(n)=n, for n=0,1;

T(n)=, for n>1.

**2-** 

**a-** *T*(*n*) = *aT*(*n/b*) + *f* (*n*)where *f*(*n*)∈Θ(*nd*), *d ≥* 0

Master Theorem: If *a < bd*, *T*(*n*) ∈ Θ(*nd*)

If *a = bd*, *T*(*n*) ∈ Θ(*nd* log *n*)

If *a > bd*, *T*(*n*) ∈ Θ(*n*log *b a* )

a=9, b=3, f(n)=n => d=1 then 9>3^1 *T*(*n*) ∈ Θ(*n*log *b a* )

**b-**

**3-** T(1)=6;T(n)=T(n-1) + n – 1.

**a-** T(n)=~~T(n-1)~~ +n-1

~~T(n-1)~~ =~~T(n-2)~~ +n-2

...

~~T(2)~~ =~~T(1)~~ +1

+ ~~T(1)~~ =15

T(n)=(n-1)+(n-2)+...+1+15=n(n-1)/2 + 15

**b-** T(n) =T(n-1) + n – 1

**c-**We can’t use master method since recurrence is not in the form of *T*(*n*) = *aT*(*n/b*) + *f* (*n*) *b>1*

**4-** 

**a-** d1=1

d2=3+1=4

d3=32+3+1

...

dn=3(n-1)+3(n-2)+...+32+3+1 this is a geometric sum

dn=(3n-1)/2 our guess

if n=2 d2=(32-1)/2=4 is true from the table above

For n=k dk=(3k-1)/2 We are assuming this is true

For n=k+1 dk?(3k+1-1)/2 we should prove

dk+1=3dk+1 => dk+1=(3(3k-1)/2)+1=(3k+1-3)/2+1=(3k+1-1)/2 this is what we were looking for therefore proven.

**b-** dn=~~3d~~~~n-1~~+1

~~3d~~~~n-1~~~~=9d~~~~n-2~~+3

~~9d~~~~n-2~~~~=27d~~~~n-3~~+9

...

~~3~~~~n-3~~~~d~~~~3~~~~=3~~~~n-2~~~~d~~~~2~~+3n-3

+ ~~3~~~~n-2~~~~d~~~~2~~=3n-1d1+3n-2

dn=3(n-1)+3(n-2)+...+32+3+1=(3n-1)/2 geometric sum

**c-** dn=3dn-1+1

dn-1=3dn-2+1 => dn=3(3dn-2+1)+1=9dn-2+3+1

dn-2=3dn-3+1 => dn=9(3dn-3+1)+3+1=27dn-3+9+3+1

...

dn=3n-1d1+3n-2+...+3+1 geometric sum

dn =(3n-1)/2

**d-**

**5-**

**a**- cn =

**b**-c(1)=1

c(2)=1

...

c(5)=2

c(6)=2

...

c(25)=3

...

cn= our guess

if n=1 c(1)= success

if n=k c(k)= assumed it is true

if n=k+1 c(k+1)?

c(k+1)=

if k+1 not power of 5 we can say

c(k+1)= proven

if k+1 power of 5 we can say

c(k+1)= (since increase on k will make our number pass a threshold of power of 5. And this will increase it is log by 1)

**d-**

def foo(n):

count=0;

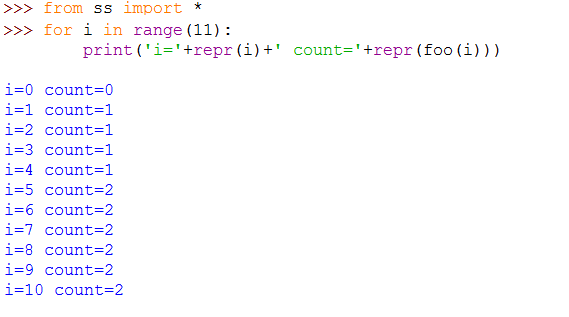
while (n>0):

n=n // 5

count=count+1

return count

for i in range(11):

print('i='+repr(i)+' count='+repr(foo(i)))

**e-** *T*(*n*) = *aT*(*n/b*) + *f* (*n*)where *f*(*n*)∈Θ(*nd*), *d ≥* 0

Master Theorem: If *a < bd*, *T*(*n*) ∈ Θ(*nd*)

If *a = bd*, *T*(*n*) ∈ Θ(*nd* log *n*)

If *a > bd*, *T*(*n*) ∈ Θ(*n*log *b a* )

cn =

a=1,b=5,d=0

1=50 => *T*(*n*) ∈ Θ(log *n*)