Name: Tian Qiu

PUID: 00265 35063

1.

(a) T(n) = 9T(n/3) + n^2 = 1 for n > 2, otherwise T(n) = 1.

a = 9, b = 3

As described in situation 2 T(n) =

(b) a = 6, b = 2

> 0 is a constant

f(n) = n^2.4 = O(

Hence, as described in situation 1, T(n) =

(c) a = 12, b = 4

f(n) = n^2 =

> 0 is a constant

Also,

for constant c = .75 < 1

so, as described in situation 3, T(n) =

2.

Base case:

T(n) = 1 < 12 <= 12\*n, for n < 4 and n is positive integer.

Assume:

For 4 <= m < n, T(m) = T(⌊2m/3⌋) + T(⌊m/4⌋) + m < 12m is right

To Prove:

T(n) = T(⌊2n/3⌋) + T(⌊n/4⌋) + n <= 12n

Prove:

Left Hand Side = 2n \* 4 + n \*3 + n = 12n <= 12n

Proved.

3.

0 cn

1 c(⌊3n/5⌋) c(⌊2n/5⌋)

2 c(3⌊3n/5⌋/5) c(2⌊3n/5⌋/5) c(3⌊2n/5⌋/5) c(2⌊3n/5⌋/5)

……

i

if level i is the longest simple path from the root to a leaf.

Because 3/5 path will be the last path to meet n < 2 which means O(1) requirement.

Then, n 🡪 (3/5)n 🡪 n(3/5)^2 🡪 …🡪1

Since, n(3/5)^k = 1 when k = .

Height of the tree is k.

We expect the solution to the recurrence to be at most the number of levels times the cost of each level.

So where c is a constant, T(n) = O(cn) = O(nlogn)