1a)

1b) (Optional lol)

2ai)

aT(n/b) + 𝚯(n^d)

2aii)

d < log\_b(a) (First case)

d = 0 < 1 = log\_2(2)

O(n^(log\_2(2)) = O(n)

2aiii)

T(n) = 2T(n/2) + 1

Trying to prove T(n) <= cn

Assume true for all m < n, especially T(n/2) <= c(n/2)

T(n) <= 2c(n/2) + 1

= cn + 1 </= cn

Fail.

Instead…

Trying to prove T(n) <= cn - d

Assume true for all m < n, especially T(n /2) <= c(n/2) - d

T(n) <= 2(c(n/2) - d) + 1

= cn - 2d + 1 <= cn - d

True when d >= 1

2bi)

def f(x, n):

if n == 0:

return 1

elif n % 2 == 0:

y = f(x, n / 2)

return y \* y

y = f(x, (n - 1) / 2)

return y \* y \* x

2bii)

Even n: T(n / 2) + 1

Master method:

d = 0, log\_b(a) = log\_2(1) = 0

d = log\_b(a)

O(n^0 lg n) = O(lg n)

(Not entirely sure if this is necessary…)

Odd n: T((n - 1) / 2) + 1

Guess O(lg n), prove T(n) <= c(lg n)

Assume true for all m < n, especially T((n-1)/2) <= c(lg ((n-1)/2))

Inductive step:

T(n) <= c(lg((n-1)/2)) + 1

= clg(n-1) - c(lg 2) + 1 <= c(lg n)

True when c >= 1/lg2

Base case:

T(3) = 2 <= c(lg 3)

True when c >= 2/lg3

Therefore O(lg n)

2c)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | - | W | E | I | R | D |
| - | 0 | 1 | 2 | 3 | 4 | 5 |
| W | 1 | 0 | 1 | 2 | 3 | 4 |
| I | 2 | 1 | 1 | 1 | 2 | 3 |
| R | 3 | 2 | 2 | 2 | 1 | 2 |
| E | 4 | 3 | 2 | 3 | 2 | 2 |
| D | 5 | 4 | 3 | 3 | 3 | 2 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| To | W | E | I | R |  | D |
| From | W |  | I | R | E | D |
| Operation | Keep | Insert | Keep | Keep | Delete | Keep |