CS325: Homework #2

1. Solve recurrence relation using three methods

$$T(n) \begin{cases} T(0) = C_1 \\ \cancel{T(1) = C_2} \\ \cancel{2(T(\frac{n}{2}))} \text{ if } n \text{ is even} \\ 2T(\frac{n}{2}) + C_3 \text{ if } n \text{ is odd} \end{cases}$$

1a. Substitution

odd: $2T(\frac{n}{2}) + Cn$

$T(n) = 2\underbrace{T(\frac{n}{2})} + Cn$ $\qquad T(\frac{n}{2}) = 2T(\frac{\frac{n}{2}}{2}) + C\cdot(\frac{n}{2})$

$2[2T(\frac{n}{4}) + C(\frac{n}{2})] + Cn$ $\qquad = 2T(\frac{n}{4}) + C\cdot(\frac{n}{2})$

$T(n) = 4\underbrace{T(\frac{n}{4})} + Cn + Cn$ $\qquad T(\frac{n}{4}) = 2T(\frac{\frac{n}{4}}{2}) + C\cdot(\frac{n}{4})$

$= 4[2T(\frac{n}{8}) + C(\frac{n}{4})] + 2Cn$ $\qquad = 2T(\frac{n}{8}) + C(\frac{n}{4})$

$T(n) = 8T(\frac{n}{8}) + Bn + Cn + Cn$

$= 2^3 T(\frac{n}{2^3}) + Cn + Cn + Cn$

k^{th} $\quad = 2^k T(\frac{n}{2^k}) + k \cdot Cn$

$\frac{n}{2^k} = 1$ $\quad T(n) = 2^{\log_2 n} T(\frac{n}{2^{\log_2 n}}) + (2^{\log_2 n} - 1) C_3$

$= nT(1) + (n-1) \cdot C_3$

$= nC_2 + (n-1) C_3$

$$\boxed{T(n) = \Theta(n)}$$

1B. master method

$T(n) = 2T(\frac{n}{2}) + C_3$

$a = 2 \quad b = 2 \quad k = 0 \quad p = 0$

$\log_2 2 = 1 > k$, $\Theta(n^{\log_b a}) = \Theta(n^{\log_2 2})$

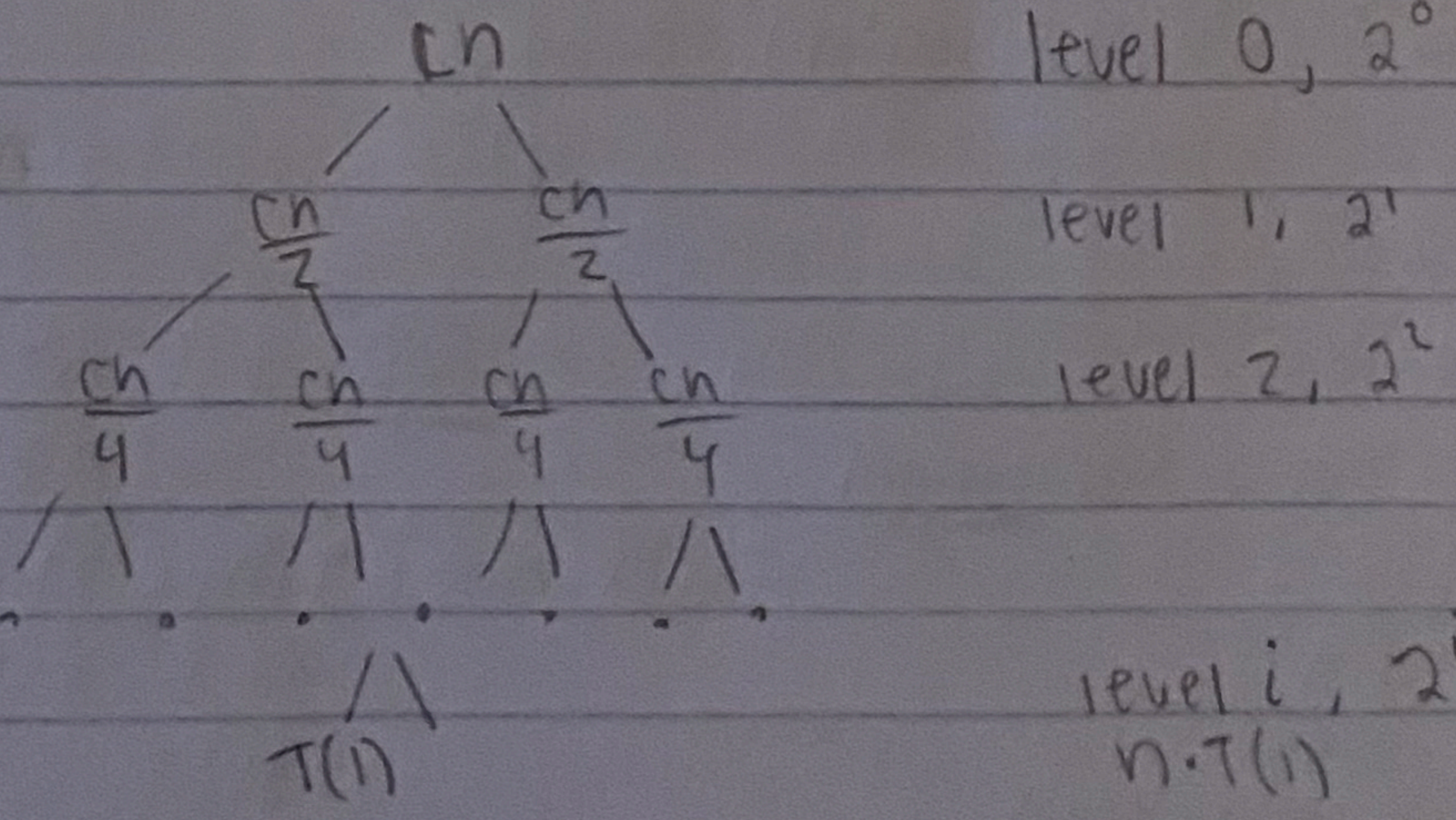
$\boxed{\text{case } 1: \Theta(n)}$

1C. tree

height $2^{\log_2 n}$

$n^{\log_2 2} = n^1$

$\Theta(n)$



2. Solve recurrence, using master theorem

2A:

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

$a = 4 \quad b = 2 \quad k = 1 \quad p = 0$

$\log_2 4 = 2 > k$

$\boxed{\text{case } 1: \Theta(n^2)}$

2B:

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

$a = 2 \quad b = 4 \quad k = 2 \quad p = 0$

$\log_4 2 = .5 < k$

$\boxed{\text{case } 3: \Theta(n^2)}$

3. Writing Pseudocode

```
function  kth element (Arr 1, Arr 2, k)
    # initialize values + 3rd arr = size of Arr1 + Arr2
loop through Arr 1 and Arr2
    if Arr 1[i] < Arr 2 [j]
        Arr3 = Arr 1
        iterate through Arr
    else    Arr 3[j] <
        Arr 3 = Arr 2
        iterate through Arr
    while more elements in Arr 1
        add to Arr3
        iterate
    while more elements in Arr 2
        add to Arr 3
        iterate
    return merged Arr(Arr3[kth element])
```