2a)

i. T(n) = 2T(n/2) + θ(1)

ii. T(n) = 7T(n/4) + θ(1)

As we don’t know the data structure holding the matrix, we can’t really assume the type of complexity it has

2b)

i. Leaves: n2; Height: log2n; Nodes: ((4log2n + 1) – 1) / 3

ii. Leaves: nlog11=n0=1; Height: n; Nodes: n

iii. Leaves: nlog53; Height: log5n; Nodes: ((3log5n + 1) – 1) / 2

iv. Leaves: nnlog11+nnlog11=n0+n0=1+1=2; Height: 2n; Nodes: 2n

2c)

T(n)=O(n)

Assume: T(k) <= ck, for all k<n

Base Case: T(1) is constant

Inductive Hypothesis: T(k) <= ck, for all k<n

Inductive Step: T(n) <= T(n/3) + T(2n/3) + n1/2

<= cn/3 + 2cn/3 + n1/2 = cn + n1/2

For any c >= 1, cn dominates n1/2, so T(n) = O(n)