Samuel Lee
ID# 20412795

Question #1

$T(n) =$

$T\left(\frac{7n}{10}\right) + T\left(\frac{1n}{4}\right) + n$

$\alpha = 1, \beta = 0$

$\left(\frac{7}{10}\right)^r + \left(\frac{1}{4}\right)^r = 1$

$\left(\frac{38}{40}\right)^r = 1 \therefore r < 1 \therefore r < \alpha \therefore \Theta(n^1 \log^0 n)$

$= \Theta(n)$

Question #2

$T(n) = 3T\left(\frac{n}{4}\right) + T\left(\frac{n}{4}\right) + n$

$\alpha = 1 \quad \beta = 0$

$3\left(\frac{1}{4}\right)^r + \left(\frac{1}{4}\right)^r = 1$

$r = 1 \therefore \Theta(n^1 \log^{\beta+1} n) = \Theta(n \log n)$

Question 3

if $n=1$, $A = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$

assume given eq. for $n=k$ is true

$$A^k = \begin{pmatrix} F_{k+1} & F_k \\ F_k & F_{k-1} \end{pmatrix}$$

to prove $A^{k+1} = \begin{pmatrix} F_{k+2} & F_{k+1} \\ F_{k+1} & F_k \end{pmatrix}$

$$A^{k+1} = A.A^k = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} F_{k+1} & F_k \\ F_k & F_{k-1} \end{pmatrix} = \begin{pmatrix} F_{k+1}+F_k & F_k+F_{k-1} \\ F_{k+1} & F_k \end{pmatrix}$$

$$= \begin{pmatrix} F_{k+2} & F_{k+1} \\ F_{k+1} & F_k \end{pmatrix} = A.A^k = A^{k+1}$$

$\therefore$ by strong induction if $A = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$, $A^n = \begin{pmatrix} F_{n+1} & F_n \\ F_n & F_{n-1} \end{pmatrix}$

Question 4:

```
Fibonacci(matrix An, int n)
        int A[2][2] = {1,1,1,0};
        if (n==0)
                return An[0][1];
        else
                An *= A;
                Fibonacci(An, n-1)
```

Question 5

input (int n, A)

priorityQueue q;

for (i=1 to n); n $O(\sqrt{\log n})$] $n\sqrt{\log n}$

q.insert(i); n

for (i=1 to n)

A[i] = q.extract(); $O(\sqrt{\log n})$] $n\sqrt{\log n}$

return A

fastest time is $O(n\sqrt{\log n})$ not $O(n \log n)$

$\therefore$ lying because $O(n \log n)$ is fastest