Solutions to problems in Homework # 3

1. Let $f(x) = \Sigma_{n=0}^{\infty} a_n x^n$ where $a_n = a_{n-1} + a_{n-2}$ for $n > 1$ and $a_0 = a_1 = 1$. Then

$$\begin{aligned} f(x) &= \Sigma_{n=2}^{\infty} a_n x^n + a_0 + a_1 x \\ &= \Sigma n = 2^{\infty}(a_{n-1} + a_{n-2})x^n + a_0 + a_1 x \\ &= x\Sigma n = 2^{\infty} a_{n-1}x^{n-1} + x^2 \Sigma n = 2^{\infty} a_{n-2}x^{n-2} + a_0 + a_1 x \\ &= x(f(x) - a_0) + x^2 f(x) + a_0 + a_1 x \end{aligned}$$

which gives us

$$f(x) = \frac{1}{1 - x - x^2}$$

Note that the roots of $1 - x - x^2$ are $(-1 + \sqrt{5})/2$ and $(-1 - \sqrt{5})/2$.
So let $f(x) = \frac{A}{(x-\alpha)} + \frac{B}{(x-\beta)}$ where $\alpha = (-1+\sqrt{5})/2$ and $\beta = (-1-\sqrt{5})/2$.
Comparing coefficients, $A = -1/\sqrt{5}$ and $B = 1/\sqrt{5}$.
Hence, finally,

$$f(x) = \frac{1/\sqrt{5}}{x - (\sqrt{5} - 1)/2} + \frac{1/\sqrt{5}}{x + (\sqrt{5} + 1)/2} \tag{1}$$

Note that $(1 + \sqrt{5})/2 = 2/(\sqrt{5} - 1)$. Substituting this we get $f(x)$ in the desired form.
Since $(-1 + \sqrt{5})/2 < 1$, $a_n \in O((1 + \sqrt{5})/2)^n)$

2. a) Size of groups=3.

$$\begin{array}{lll} T(n) & \leq kn & n < s \\ T(n) & = T(n/3) + T(2n/3) + kn & n > s \end{array}$$

for some s.

I will show that $T(n) = \Theta(nlogn)$. Hypothesis: $T(n) \leq cn$ for some constant c that depends on s. The base case is easily checked. Consider $T(n+1)$.

$$\begin{aligned} T(n+1) &\leq 1/3c(n+1)log(n+1)/3 + 2(n+1)c/3log2(n+1)/3 + k(n+1) \\ &= c(n+1)log(n+1) + (n+1)[2c/3 - clog3 + k] \end{aligned}$$

Hence if

$$[2c/3 - clog3 + k] <= 0$$

then $T(n+1) \leq c(n+1)log(n+1)$. This happens when $k \geq \frac{c}{log3-2/3}$.

With a similar argument, one can show that $T(n) \geq hn$ for some constant h.

b) Size of groups = 7

$$\begin{array}{rcl} T(n) \leq & kn & n < s \\ T(n) = & T(n/7) + T(5n/7) + kn & n > s \end{array}$$

We will show that $T(n) \leq cn$ for some constant c. The base case is easily checked. Conside $T(n+1)$.

$$\begin{aligned} T(n+1) &\leq c(n+1)/7 + 5c(n+1)/7 + k(n+1) \\ &= 6c(n+1)/7 + k(n+1) \end{aligned}$$

Hence when $c > 7k$ the RHS $\leq c(n+1)$.

3. Let the list of sorted file sizes be $s_1 \leq s_2 \ldots \leq s_n$. The algorithm uses a queue Q (initialized to be empty) to store the sizes of the intermediate files.