1 Homework 2

Name: Kangdong Yuan

1.1 probelm1

a).I did not work in a group.
b).I did not consult without anyone my group members
c).I did not consult any non-class materials.

1.2 problem2

a).$f = \Omega(g)$
both f and g are the power of n, but $1.5 > 1.3$, so $\lim_{n\to\infty} \left(\frac{f(n)}{g(n)}\right) = \infty$.
b).$f = \Theta(g)$
because $f(n) = 0.5 * 2^n = O(2^n) = g(n)$ for $\lim_{n\to\infty} \left(\frac{f(n)}{g(n)}\right) = 0.5$.
c).$f = \Omega(g)$
if base of log is 2, for $n \geq 2.23$, $1.3log(n) > 1.5$, so the $n^{1.3log(n)} > n^{1.5}$. $\lim_{n\to\infty} \left(\frac{f(n)}{g(n)}\right) = \infty$.
d). $f = \Omega(g)$
the $f = \Theta(3^n)$ and $g = \Theta(2^n)$. The base of n are $3 > 2$. So for $\lim_{n\to\infty} \left(\frac{f(n)}{g(n)}\right) = \infty$.
e).$f = O(g)$
take log on f and g, $log(f) = 100log(log(n)) < 0.1log(n) = log(g)$, so $\lim_{n\to\infty} \left(\frac{f(n)}{g(n)}\right) = 0$.
f). $f = \Omega(g)$
take log for f and g, $log(f(n)) = log(n) > loglog(n) * log(log(n))$, so $\lim_{n\to\infty} \left(\frac{f(n)}{g(n)}\right) = \infty$.
g). $f = O(g)$
take log for f and g, $log(f(n)) = log(n) < \prod_{i=0}^{n} log\,(i) = log(g(n))$,
so $\lim_{n\to\infty} \left(\frac{f(n)}{g(n)}\right) = 0$.
h).$f = O(g)$
$f(n) = nlog(e)$, for $n > e$, the $log(e) < log(n)$.
So for $n > 1$, $f(n) = nlog(e) < nlog(n) = g(n)$.
Thus $\lim_{n\to\infty} \left(\frac{f(n)}{g(n)}\right) = 0$.
i).$f = \Theta(g)$
$f(n) = n + log(n) = O(n) = n + (log(n))^2 = g(n)$ for all $n > 1$.
Thus $\lim_{n\to\infty} \left(\frac{f(n)}{g(n)}\right) = c,\ c\ is\ a\ constant\ number$

j).$f = \Theta(g)$
$f(n) = 5n + \sqrt{n} = O(n) = log(n) + n = g(n)$ for all $n > 1$.
Thus $\lim_{n\to\infty}\left(\frac{f(n)}{g(n)}\right) = c,\ c\ is\ a\ constant\ number$

1.3 problem3

a). There are $n+1$ items in polynomial. The sum of $n+1$ items take n steps. The multiplications in i items is $i+1$, the total multiplications is $\sum_{i=0}^{n} i+1 = 1 + \frac{n^2+3n}{2}$.
Adding all sum and multiplications together, $\frac{n^2+3n}{2} + n + 1 = O(n^2)$
The time complexity in this polynomials is $O(n^2)$
b).first, we prove the base case, for $n = 1$, $z_n = a_1x_0 + a_0 = p_n(x_0)$.
we need to prove the case for $n+1$ is also true. $p_{n+1}(x_0) = p_n(x_0) + a_{n+1}x^{n+1} = z_n + a_{n+1}x^{n+1} = z_{n+1},\ because\ z_n = p_n(x_0)$
so, we can infer that $z_n = p_n(x_0)$, for all n, and the Horner's rule output the $p_n(x_0)$.
c). In the Horners algorithm, it go through the loop form $n-1$ to 0. In each loop, it has two steps. $\sum_{i=n-1}^{0} 2 = O(n)$
The time complexity of Horners algorithm is $O(n)$

1.4 problem4

a). $T(n) = 2T(\frac{n}{2}) + \sqrt{n}$
The branching factor is 2, and the number of subproblems at depth k are 2^k, the size of subproblems at depths k is $2^k * \sqrt{\frac{n}{2^k}}$. The total depth is $\frac{n}{2^k} = 1,\ k = log_2(n)$. The total running time is $\sum_{k=1}^{log(n)} \sqrt{\frac{n}{2^k}} = \Theta(n)$

b).The branching factor is 2, and the number of subproblems at depth k are 2^k, the size of subproblems at depths k is 2^k. The total depth is $\frac{n}{3^k} = 1,\ k = log_3(n)$. The running time is $\sum_{k=0}^{log_3(n)} 2^k = \Theta(n^{log_3^2})$.

c).The branching factor is 5, and the number of subproblems at depth k are 5^k, the size of subproblems at depths k is $\frac{5^k*n}{4^k}$. The total depth is $\frac{n}{4^k} = 1,\ k = log_4(n)$. The running time is $\sum_{k=0}^{log_4(n)} \frac{5^k \cdot n}{4^k} = \Theta(n^{log_4^5})$.

d).The branching factor is 7, and the number of subproblems at depth k are 7^k, the size of subproblems at depths k is n. The total depth is $\frac{n}{7^k} = 1,\ k = log_7(n)$. The running time is $\sum_{k=0}^{log_7(n)} n\ = \Theta(n * log_7^n)$.

e).The branching factor is 9, and the number of subproblems at depth k are 9^k, the size of subproblems at depths k is n^2. The total depth is $\frac{n}{3^k} = 1,\ k = log_3(n)$. The running time is $\sum_{k=0}^{log_3(n)} n^2 = \Theta(n^2 * log_3^n)$.