Recommended HW

Problem 1:

a. $T(n) = T(i) + T(n-i-1) + 1$
b. for all value $i = 0, 1, ..., n-1$
c. $T(1) = T(0) + T(0) + 1$
d. $T(2) = T(0) + T(1) + 1, T(2) = T(1) + T(0) + 1$
e. $T(3) = T(0) + T(2) + 1, T(3) = T(1) + T(1) + 1, T(3) = T(2) + T(0) + 1$
f. $T(4) = T(0) + T(3) + 1, T(4) = T(1) + T(2) + 1, T(4) = T(2) + T(1) + 1, T(4) = T(3) + T(0) + 1$
g. ...
h. $nT(n) = 2(T(0) + T(1) + ... + T(n-2) + T(n-1)) + n$
i. $nT(n) = (n+1)T(n-1) + 1$
j. $\frac{T(n)}{n+1} = \frac{T(n-1)}{n} + \frac{1}{n(n+1)} \to \frac{1}{2} + \frac{1}{2\times3} + ... + \frac{1}{n(n-1)} + \frac{1}{n(n+1)}$ where the nominator sums up to n then $T(n) = n$

Problem 2:

$2^{2^{n+1}}$

$2^{2^{n+1}}$	
2^{2^n}	
$(n+1)!$	
$n!$	
$n2^n$	
e^n	
2^n	
$\frac{3}{2}^n$	
$log(n)!$	
$n^{loglog(n)}$	$log(n)^{log(n)}$

n^3	
n^2	$4^{log(n)}$
$nlog(n)$	$log(n!)$
$2^{log(n)}$	n
$\sqrt{2}^{log(n)}$	
$2^{\sqrt{2log(n)}}$	
$log^2(n)$	
$ln(n)$	
$\sqrt{log(n)}$	
$ln(ln(n))$	
2^{log^*n}	
log^*n	$log^*(log(n))$
$log(log^*(n))$	
1	$n^{1/log(n)}$

Problem 3:

a. $T(n) = T(n/2) + n^3$
   a. At each node level we have $(n/2^k)^3 = n^3/8k^3$ total cost at each depth k.
   b. The tree depth is $log_2(n) + 1$ (root)
   c. In the last level we have $n^3 \sum_{k=0}^{log(n)} \frac{1}{8^k}$ nodes in total.
   d. In total of all recursive levels we have $n^3 \cdot \frac{1}{1-1/8} = 8/7 \cdot n^3 = O(n^3)$ since the total is a decreasing geometric sequence.

b. $T(n) = 4T(n/3) + n$
   a. At each node level we have $n(4/3)^k$ total cost at each depth k.
   b. The tree depth is $log_3 n + 1(root)$
   c. In the last level we have $4^{log_3 n} = n^{log_3 4}$ nodes in total.
   d. In total of all recursive levels we have $\sum_{k=0}^{log_3 n} (\frac{4}{3})^k n + \theta(n^{log_3 4}) = n \cdot \frac{(4/3)^{log_3 n} - 1}{(4/3) - 1} + \theta(n^{log_3 4}) = 3n(4^{log_3 n})3^{-log_3 n} + \theta(n^{log_3 4}) = 3n \cdot n^{log_3 4} \cdot n^{-log_3 3} + \theta(n^{log_3 4}) = 3n^{log_3 4} + \theta(n^{log_3 4}) = \theta(n^{log_3 4})$

e.

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

a. At each node level we have $n \cdot 2^k$ total cost at each depth k.

b. The tree depth is $log_2(n) + 1$(root).

c. In the last level we have $4^{log_2 n} = n^2$ nodes in total.

d. In total of all recursive levels we have $\sum_{k=0}^{log_2 n} 2^k n + \theta(n^2) = n \cdot \frac{2^k - 1}{2-1} + \theta(n^2) = n \cdot 2^k - 1 + \theta(n^2) = \theta(n^2)$

d. $T(n) = 3T(n-1) + 1$

a. At each node level we have 3^k total cost at each depth k.

b. The tree depth is $log_2(n) + 1$(root).

c. In the last level we have 3^n nodes in total.

d. In total of all recursive levels we have $\sum_{k=0}^{n} 3^k n + \theta(3^n) = n \cdot \frac{3^k - 1}{3-1} + \theta(3^n) = n \cdot (3^k - 1)/2 + \theta(3^n) = \theta(n \cdot 3^n)$

Problem 4:

$T(n) = T(\alpha) + T((1-\alpha)n) + \theta(n)$ where $0 < \alpha < 1$.