

COT5405: Homework 2 (Spring 2015)

1. **(35 points) Recurrence Complexity Analysis:** For recursive divide-and-conquer algorithms, we use $T(n)$ to represent the complexity of problem with size n . For each of the following recurrences,

- Draw the recursion tree
- Indicate the depth of the tree and the total number of leaves
- Calculate the asymptotic total cost assigned to the nodes in the tree. That is, estimate the big-Theta expression for $T(n)$. **Note: you can represent the solution either by n , or by k where $k = \log_3 n$ for the first three questions.**

Assume $T(n)$ is equal to 1 for $n < 3$, and algorithms stop recursive operation when $n < 3$.

1 $T(n) = 5T(n/3) + n$

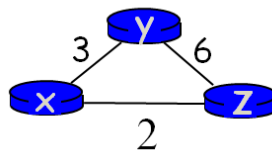
2 $T(n) = 9T(n/3) + n^2$

3 $T(n) = 2T(n/3) + n$

4 $T(n) = 2T(n - 2) + 1$

For simplicity, assume for the (1)-(3) questions, $n = 3^k$ where k is an integer; and for (4) question, n is an odd integer ($n=2k+1$).

2. **(30 points) Distance Vector Protocol:** Please study the example in slides '06bellman-ford.ppt', Page 16,17. Use the similar way to illustrate how the three nodes exchange and update their distance table to converge, when the link cost between the three nodes are shown below.



3. **(35 points) Ford-Fulkerson Algorithm:** Please study the demo example in slides '07demo-maxflow.ppt'. I used the same graph with the same capacity for this question. Suppose the following graph shows the flow values (blue numbers) on all edges at current step in running the Ford-Fulkerson algorithm.

(1) Draw the corresponding Residual graph G_f and derive the overall Flow value.

(2) Suppose the next augmenting path P on the Residual graph we find is Path $s \rightarrow 2 \rightarrow 4 \rightarrow 5 \rightarrow t$. Derive the new overall Flow value. Draw the updated graph G with the new flow values on all edges.

(3) Then draw the new updated corresponding Residual graph G_f

