Homework assignment 2, due Tues Oct. 9 before class (one week later for FEEDS students)

–Look up course webpage under "Homeworks," specifically for
late submission policy which you must follow strictly – otherwise you could lose all points;
–Cheating policy and bonus policy under "about homeworks."
–Please make sure to include your code when you submit your homework, or risk losing points.
Most answers should contain at least the following 4 parts. (Certain problems may require more, and a very few, less).
- The idea of the algorithm – you could choose to illustrate this with an example. Mention familiar design techniques used, if any.
- A pseudocode of the algorithm. Any segment of code that takes constant time may be abbreviated by one instruction. Any algorithm done in class or in the assigned readings can be quoted. All else must be explained.
- A proof of correctness: argue why the desired output is produced when the algorithm halts, and that it does halt.
- An analysis of the worst case time complexity, in terms of the relevant input size parameters.

1. **(Compulsory, 20 points)**

   (a) Design the most efficient algorithm you can for the following problem.
   Input: A connected, undirected graph $G = (V, E)$, with a distinct positive integer $k(v)$ attached to every vertex $v \in V$. The graph additionally has the property that every vertex - except one namely *root* - is adjacent to at least one vertex with a larger number. I.e, for every $v \in V$ except *root*, there is a $u \in V$ such that $k(u) > k(v)$ and $(u, v) \in E$.
   Output: A spanning tree $T = (V, E', root)$ for G that satisfies the following: the number at every vertex in T is greater than the number stored at all of its children, i.e, T is a kind of heap, but need not be balanced and could allow any number of children (possibly more than two) for each vertex.

   (b) Design a constant $O(1)$ time algorithm for the following problem.
   Input: A connected, undirected graph $G = (V, E)$, with a distinct

positive integer $w(e)$ attached to every edge $e \in E$.
Output: Yes, if and only if the minimum weight spanning tree for G is *unique*.

2. **(Compulsory 20 points)** Design the most efficient algorithm you can for the following problem.
Input: Positive integer weights w_{ij} associated with every interval (i, j), where $1 \leq i < j \leq n; i, j \in N$ (i.e, the endpoints of the intervals are the positive integers i and j between 1 and n).
Output: A minimum weight nonoverlapping cover of the entire interval $(1, n)$. I.e, a set of intervals C that are pairwise nonintersecting (except at end points); every interval $(i, i+1)$ for $1 \leq i < n$ properly intersects (is contained in) atleast one of the intervals in C; and total weight of intervals in C is minimum. *Hint: Use dynamic programming.*

3. **(Compulsory, 20 points)** Problem 16-1 in book. Include the following question (c'): Give the most general property you can such that: for any set of coin denominations (which include a penny) satisfying your property, the greedy algorithm would always yield the optimal solution.

4. **(Compulsory, 20 points)** 17.3-2, 17.3-3, 17.3-4, 17.3-7. Only 17.3-7 requires the first three 'algorithm design' parts given under instructions above. The other three just require the analysis of amortized time complexity.

5. **(Optional 20 Bonus pts–see 'bonuses and incentives' in webpage)** Let G be a connected undirected graph with two positive values

   a_{ij} and b_{ij} associated with each edge (i, j). For a spanning tree T of G, let $A(T)$ (resp. $B(T)$) denote the sum of the a_{ij}'s (resp. b_{ij}'s) over the edges of T. Find a spanning tree T that minimizes $A(T)/B(T)$.

   Big Hint. Use binary search and the following thoughts (prove any claims). Let t be the optimal value of the ratio. If you "guess" t at each stage of the search and solve the MST problem with weights $a_{ij} - tb_{ij}$, then the weight of the MST is positive (resp. negative) iff the guessed t is too small (resp. too large). Then the real line is the union of open intervals, each of which corresponds to a particular tree being picked. The boundaries between the intervals are the points t for which $a_{ij} - bt_{ij} = a_{kl} - tb_{kl}$ for some i, j, k, l (not obvious!).