CE-1
August 2010 QE

1. **(20 points: Asymptotic analysis)**

   Express the complexity of the function DIV-AND-CONQUER (shown below) as a recurrence. Assume that the asymptotic time complexity of the functions Decompose(p,q) and Combine(p,q) are D(q-p+1) and C(q-p+1), respectively.

```
DIV-AND-CONQUER(low,hi)
Begin
      range=hi-low+1;
      quarter = low + range/4 ;
      half = low + range/2 ;
      threequarters = low + 3*range/4 ;

      If (range < 1000)
            Return (range);
      Else
            Decompose(low,hi);
            DIV-AND-CONQUER (low,quarter-1);
            DIV-AND-CONQUER (quarter,half-1);
            DIV-AND-CONQUER (half,threequarters-1);
            DIV-AND-CONQUER (threequarters, hi);
            Combine(low,hi);
      Endif
End
```

   If $D(n) = \Theta(\lg n)$ and $C(n) = \Theta(\lg(n^2))$, obtain asymptotic bounds for the time complexity of DIV-AND-CONQUER. Assume that N is power of 4 and that the top level call is DIV-AND-CONQUER(1,N).

2. **(25 points: Graph Algorithms : Descent-only Shortest Paths)**
   A number of houses are located on a hill-slope such that given any pair of houses h_1 and h_2, either h_1 is at a higher altitude or h_2 is at a higher altitude. Assume that the set of houses is H. There are a set of roads **R** that run in straight lines between selected pairs of houses. Briefly describe an efficient algorithm to find the shortest distance from a given house ***h*** to all other houses such that **only descending roads** (i.e., roads from a house at a higher altitude to a house at a lower altitude) are traversed. (If your algorithm uses known graph-algorithms as components, you may simply name them without describing them in detail.) What is the asymptotic time complexity of your algorithm?

3. **(25 points: Greedy Algorithms)**
   Given a weighted, undirected graph G(V,E) describe an efficient **greedy** algorithm to identify at least |V|/2 edges of the graph such that all identified edges are on the graph's minimum-spanning-tree. What is the asymptotic complexity of your algorithm? Hint: A naïve solution is to run Kruskal's algorithm or Prim's algorithm and stopping after |V|/2 edges are identified. Your algorithm must be better.

4. **(30 points: Complexity theory: Scheduling with Constraints (SWC))**

   M managers must attend one or more of T meetings. Each manager m is represented by the set of meetings he/she has to attend, which is a subset of T. Each meeting must be scheduled on one of K days such that no two meetings t1 and t2 must be scheduled on the same day d if a manager has to attend both meetings (i.e., a manager can attend only one meeting a day). Further, there are some meetings whose dates are fixed because of an external constraint. These constraints are expressed as a set of constraints FIXED = { (t,k) | t in T and 1<=k<=K} to indicate the specific day k on which the meeting t has to be scheduled.

   The scheduling decision problem SWC (M,T,K,FIXED) evaluates whether there exists any schedule that maps meetings to days such that (1) each manager can attend all of his/her meetings without attending more than one meeting a day AND (2) the fixed meetings are held on the specified days.

(a) Prove that SWC is in NP.

(b) Prove that SWC is NP-Complete. (Hint: GRAPH-COLOR(G,k), a decision problem that evaluates whether a given graph can be k-colored such that no two adjacent vertices share the same color, is known to be NP-Complete.)