CE-1
August 2017 QE

When an English algorithm description is requested, you may provide pseudo-code as well, for clarity; however, your English description must be complete. Pseudo-code may be ignored at our discretion. We recommend saving at least 30 minutes for the final two questions if you seek a full score.

1. **(25 points)** For each of the following code fragments, give a Θ bound describing the asymptotic number of basic steps executed in terms of positive integer n. Justify your answer carefully. For the second fragment, state the value computed.

   (a)
   ```
   sum = 1;
   for i = n*n downto 1
       for j = 1 to i
           sum = sum + 1;
   ```
   (b)
   ```
   foo(x, n)
       if (n==0) return 1;
       if (n==1) return x;
       if (n%2==0)
           then return foo(x*x, n/2));
           else return x * foo(x*x, n/2));
   ```
   Suppose the call foo(5,n) is executed. Here, % denotes the modulo operation, and integer division discarding the remainder is represented by /. For your asymptotic bound, consider n an exact power of 2.

2. **(25 points)** Given as input an array $A[1..n]$ of integers, describe in English a worst-case quadratic algorithm to produce a matrix $B[1..n, 1..n]$ of integers such that $B[i, j]$ equals the sum of the integers $A[i..j]$ in the array A indexed between i and j, with $B[i, j]$ equal to 0 when $i > j$. Argue briefly that your algorithm is correct and quadratic in the worst case.

3. **(20 points)** Carefully define the binary relation "polynomially reduces" ($\preccurlyeq_p$) and prove that this relation is transitive. Include careful description of the domain of this relation.

4. **(15 points)** Consider the Bellman-Ford algorithm, run on a graph of n vertices with real edge weights, with no negative cycles. After $n-1-k$ iterations of the algorithm's main loop, for some positive integer k, what is the largest number of vertices that can still have incorrect shortest-path distance bounds? For half credit, justify your answer with convincing example graphs and an explanation. For full credit, justify your answer with a careful proof.

5. **(15 points)** Show that the problem HITTINGSET is $\mathcal{NP}$-hard given that the problem VERTEXCOVER is $\mathcal{NP}$-hard. Five of the points for crisply elaborating what must be exhibited.

   VERTEXCOVER asks whether there is a size k vertex cover of a graph G. A vertex cover is a subset of vertices such that every edge starts or ends at a vertex in the subset.

   HITTINGSET considers a sequence of sets $S_1, ..., S_m$ and a positive integer x and seeks a subset H of the domain $\bigcup_i S_i$ such that for every i, $H \cap S_i$ is non-empty.

Write in Exam Book Only