

# Theory Qualifier Exam

Your number: \_\_\_\_\_

**Time limit: 2.5 hours. Use only the notes supplied by the Department**

FALL 2010, CS DEPARTMENT, IIT

For every question, please write your answer in a clean and concise way. Use additional pages, start a new page with each problem and write only on one side of the paper.

If you are asked to write an algorithm for a question, you have to write the **pseudo-code** of your algorithm and also put explanations about your pseudo-code. Also show correctness and estimate the running time.

Use procedures if you want – marking clearly what the parameters are and what they do, and with what running time in terms of its parameters. Unless the procedures are from the textbooks, write pseudocode for the procedures.

1. Present an algorithm with running time  $O(n)$  for the following problem: Given array  $A[1..n]$ , find entries  $1 \leq i < j \leq n$  such that  $A[j] - A[i]$  is maximum among all such pairs  $i < j$ . As an example, if  $A = \langle 9, 1, 5 \rangle$ , output  $i = 2$  and  $j = 3$ . A  $O(n \log n)$  algorithm is worth partial credit.
2. Given a directed graph  $G = (V, E)$  we define the graph  $G^2 = (V, E^2)$ , such that  $(u, w) \in E^2$  if and only if for some  $v \in V$ , both  $(u, v) \in E$  and  $(v, w) \in E$ . That is,  $G^2$  contains an edge between  $u$  and  $w$  whenever  $G$  contains a path with exactly two edges between  $u$  and  $v$ . Describe an efficient algorithm for computing the adjacency matrix of  $G^2$  given the adjacency matrix of  $G$ . Present the pseudocode and analyze the running time in terms of  $|V|$  and  $|E|$ .
3. Prove that  $A_{TM}$  is not mapping reducible to  $E_{TM}$ .
4. Factoring an integer  $i$  means finding the smallest factor of  $i$  other than 1. As an example, if  $i = 12$ , factoring 12 produces the number 2. Also, if  $i$  is prime, factoring  $i$  produces  $i$ .

Prove that, if  $P = NP$ , there is a polynomial-time algorithm for factoring. You can assume the existence of polynomial-time algorithms for integer addition, integer multiplication, and integer division. (Note:  $NP$  is a class of *languages*, and factoring is a *function*.)