gcd(a, b) = gcd(b, a mod b)

Shaurya Gomber

160101086

Major Exam 1*

Algorithms

Spring 2018 CS204@IITG

- This is a closed book exam.
- The solutions to ⋄ marked ones were presented in lectures.
- In solving problems, you can use the Theorems and Lemmas proved in class as they are i.e., without re-proving them (of course, unless the question is marked with either ⋄ or explicitly asks you to start from primitive definitions).
- If something is missing/incorrect in a problem description, clearly mention the assumption with your solution.
- Every solution must be supported by a formal proof of correctness and analysis to get any credit.
- Be precise; there is no credit for being unnecessarily verbose.
- Do write both the name and roll number on your answer sheets.
- Unless specified otherwise, assume word-RAM model of computation.
- Negative marking is in place for counterintutive solutions, algorithms, or proofs. Hence, it is risky to fill papers with no obvious necessity.

$c_1 A' \mid c_2 A'$

$c_3 A' \mid c_4 A'$

(1) Let A be a matrix of size $n \times n$ that has the following *recursive* structure: when it is partitioned into north-east (NE), north-west (NW), south-east (SE) and south-west (SW) submatrices of equal sizes (each is of order $\frac{n}{2} \times \frac{n}{2}$), $NW = c_1A', NE = c_2A', SW = c_3A', SE = c_4A'$ where A' is matrix of same type as A (defined recursively) and c_1, c_2, c_3, c_4 are integers. Devise an efficient algorithm to compute AV where V is a column vector (of order $n \times 1$) of positive integers. Assume n is a power of 2.

(2) Give a recurrence relation for computing the greatest common divisor of two positive integers. Use this recurrence in devising a divide-conquer-combine based efficient algorithm to compute the gcd of any two positive integers. Assume RAM model of computaion.

RAM

(3) ⋄ Using the $O(n \lg n)$ worst-case time algorithm presented in lectures, compute the DFT of polynomial $(1 - 3x + 2x^2)$. Do detail all the steps in applying that algorithm to the given problem instance.

(4) Devise an efficient greedy algorithm for the following problem and prove its correctness using the exchange argument:

Let S be a set of n items. For $i = 1, \ldots, n$, the i^{th} item of S is associated with a nonnegative volume w_i and a value v_i. Further, a bag B that can carry up to volume W is given. Find the mix of items that can fit in B so that the total value of the items picked into B is maximized. For any item $I \in S$, you are allowed to pick parts (fraction) of I as well.

(5) Let S be a set of n distinct positive integers, each associated with a positive weight. Devise an efficient algorithm to determine the element x in S so that $\sum_{y \in S, y<x} weight(y) < \frac{\|S\|}{2}$ and $\sum_{y \in S, y>x} weight(y) \leq \frac{\|S\|}{2}$. Here, $\|S\| = \sum_{y \in S} weight(y)$ and $\|S\|$ is assumed to be even.

(6) Devise an algorithm to compute the convex hull of a set S of n points in $\mathbb{R}^2$. Your algorithm must be based on divide-and-conquer paradigm and it should take $O(n \lg n)$ time and $O(n)$ work-space in the worst-case.

*Prepared by R. Inkulu, Department of Computer Science, IIT Guwahati, India. http://www.iitg.ac.in/rinkulu/

n positive integers, all have a weight)

W

volume →

value →

1 2 3 4

$< \frac{|S|}{2}$ $\leq \frac{S}{2}$

|S| to be