

NAME: \_\_\_\_\_

## ALGORITHMS QUALIFYING EXAM

This exam is open books & notes  
and *closed* neighbors & calculators.

The *upper bound* on exam time is 3 hours.

Please put all your work on the exam paper.

Please write your name on each page.

(Partial credit will only be given if your work is shown.)

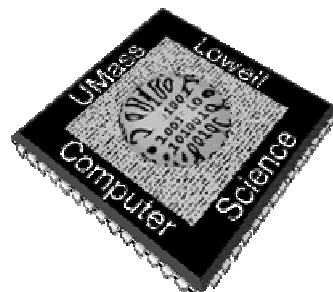
If you use pseudocode from a reference, such as the Cormen *et al.* textbook, you can just give the page reference instead of writing that pseudocode on your exam paper.

All algorithms referred to here use a sequential model of computation, not a parallel one.

If asked to provide an algorithm, your solution should contain:

- 1) pseudocode that follows the conventions of the algorithms book by Cormen, et al.;
- 2) correctness justification;
- 3) asymptotic complexity analysis (worst-case running time upper bound unless otherwise specified).

**Good luck!**



NAME: \_\_\_\_\_

**Part 1**

(86 points)

**Solve all of problems 1-6**

**1: (15 points) Boolean Expressions**

**Given:** A Boolean expression in conjunctive normal form such that each clause contains exactly three literals.

**Question:** Does there exist a satisfying assignment for the expression such that exactly one literal in each clause has the value TRUE?

***Either prove that this problem is NP-complete or provide a polynomial-time algorithm for solving it.***

***NAME:*** \_\_\_\_\_

NAME: \_\_\_\_\_

## 2: (14 points) Bitonic Shortest Paths

**A sequence is bitonic if it monotonically increases and then monotonically decreases, or if it can be circularly shifted to monotonically increase and then monotonically decrease. For example, the sequences  $\langle 1,4,6,8,3,-2 \rangle$ ,  $\langle 9,2,-4,-10,-5 \rangle$ , and  $\langle 1,2,3,4 \rangle$  are bitonic, but  $\langle 1,3,12,4,2,10 \rangle$  is not.**

**Suppose that we are given a directed graph  $G = (V, E)$  with weight function  $w : E \rightarrow R$ , and we wish to find single-source shortest paths from a source vertex  $s$ . We are given one additional piece of information: for each vertex  $v \in V$ , the weights of the edges along any shortest path from  $s$  to  $v$  form a bitonic sequence.**

**Give the most efficient algorithm you can to solve this problem, and analyze its running time.**

**NAME:** \_\_\_\_\_

NAME: \_\_\_\_\_

### 3: (14 points) Graph Reachability

Let  $G = (V, E)$  be a directed graph in which each vertex  $u \in V$  is labeled with a unique integer  $L(u)$  from the set  $\{1, 2, \dots, |V|\}$ . For each vertex  $u \in V$ , let  $R(u) = \{v \in V : u \rightsquigarrow v\}$  be the set of vertices that are reachable from  $u$  by following a path along edges of  $G$ . Define  $\min(u)$  to be the vertex in  $R(u)$  whose label is minimum, i.e.,  $\min(u)$  is the vertex  $v$  such that  $L(u) = \min\{L(w) : w \in R(u)\}$ . Give an  $O(V + E)$ -time algorithm that computes  $\min(u)$  for all vertices  $u \in V$ .

**NAME:** \_\_\_\_\_

*this page left blank for work*

**NAME:** \_\_\_\_\_

#### **4: (14 points) Scheduling to Maximize Profit**

Suppose you have one machine and a set of  $n$  jobs  $a_1, a_2, \dots, a_n$  to process on that machine. Each job  $a_j$  has a processing time  $t_j$ , a profit  $p_j$ , and a deadline  $d_j$ . The machine can process only one job at a time, and job  $a_j$  must run uninterruptedly for  $t_j$  consecutive time units. If job  $a_j$  is completed by its deadline  $d_j$ , you receive a profit  $p_j$ ; if it is completed before its deadline, your profit goes up by a certain base amount  $b$  times the number of time units gained; but if it is completed after its deadline, you receive a “negative profit” (i.e., you are penalized) and your profit goes down by the base amount  $b$  times the number of time units lost. Give an algorithm to find the schedule that obtains the maximum amount of profit, assuming that all processing times are integers between 1 and  $n$ . What is the running time of your algorithm?

***NAME:*** \_\_\_\_\_

*this page left blank for work*

NAME: \_\_\_\_\_

### 5: (14 points) Binary Search

Binary search of a sorted array takes logarithmic search time, but the time to insert a new element is linear in the size of the array. We can improve the time for insertion by keeping several sorted arrays.

Specifically, suppose that we wish to support SEARCH, and INSERT, and DELETE on a set of  $n$  elements. Let  $k = \lceil \lg(n+1) \rceil$ , and let the binary representation of  $n$  be  $\langle n_{k-1}, n_{k-2}, \dots, n_0 \rangle$ . We have  $k$  sorted arrays  $A_0, A_1, \dots, A_{k-1}$ , where  $i = 0, 1, \dots, k-1$ . The length of array  $A_i$  is  $2^i$ . Each array is either full or empty, depending on whether  $n_i = 1$  or  $n_i = 0$ , respectively. The total number of elements held in all  $k$  arrays is, therefore,  $\sum_{i=0}^{k-1} n_i 2^i = n$ . Although each individual array is sorted, there is no particular relationship between elements in different arrays.

- a. Describe how to perform the SEARCH operation for this data structure. Analyze its worst-case running time.
- b. Describe how to INSERT a new element into this data structure. Analyze its worst-case and amortized running times.
- c. Describe how to perform the DELETE operation for this data structure. Analyze its worst-case running time.

***NAME:*** \_\_\_\_\_

*this page left blank for work*

**NAME:** \_\_\_\_\_

**6: (14 points) Strings**

Given a string  $T$  represented as an array of characters  $T[1 \dots n]$ , define a tandem substring  $\alpha$  of a base  $\beta$  to be a substring of  $T$  consisting of at least one consecutive copy of  $\beta$ . A *maximal* tandem substring is a tandem substring that cannot be extended either to the left or to the right.

Design an efficient algorithm that accepts, as input, a string  $T$  and a base  $\beta$  and finds the longest maximal tandem substring of  $T$  using  $\beta$ . The algorithm should output this substring, its starting index in  $T$  and the number of copies of  $\beta$  that it contains.

For example, let  $T = \text{abcabcababcab}$  and  $\beta = \text{abcab}$ . The longest maximal tandem substring of  $T$  using  $\beta$  is  $\text{abcababcab}$ , which contains 2 consecutive copies of  $\beta$  and begins at position 4 in  $T$ .

**NAME:** \_\_\_\_\_

*this page left blank for work*

NAME: \_\_\_\_\_

**Part 2**  
(14 points)

**Solve one of problems 7-9.**

**7: (14 points) Linear Programming**

**A nut company sells 3 assortments of mixed nuts. The types of nuts are: almonds, pecans, cashews and walnuts.**

<b>Name of Assortment</b>	<b>Requirements</b>	<b>Selling Price per Pound</b>
Regular	Not more than 20% cashews. Not less than 40% walnuts. Not more than 25% pecans.	59 cents
Deluxe	Not more than 35% cashews. Not less than 25% almonds.	69 cents
Blue Ribbon	Between 30-50% cashews. Not less than 30% almonds	85 cents

**The cost and availability of the nuts is given by:**

<b>Type of Nut</b>	<b>Cost per Pound</b>	<b>Number of Pounds Available Each Week</b>
Almonds	25 cents	2000
Pecans	35 cents	4000
Cashews	50 cents	5000
Walnuts	30 cents	3000

**The goal is to determine how many pounds of each type of nut to put into each assortment each week in order to maximize total net weekly profit.**

**Formulate a linear program that solves this problem. That is, describe its objective function, constraints and bounds on variables. Explain the meaning of each variable.**

**Note: You *do not* need to solve the linear program.**

***NAME:*** \_\_\_\_\_

*this page left blank for work*

**NAME:** \_\_\_\_\_

**8: (14 points) Computational Geometry**

A point  $p$  in the plane *dominates* another point  $q$  if both the  $x$  and  $y$  coordinates of  $p$  are greater than or equal to those of  $q$ . For example, point  $(1,3)$  does not dominate point  $(2,2)$ . However, point  $(3,3)$  dominates both points  $(1,3)$  and  $(2,2)$ . A point  $p$  is a *maximal* point in a given set of points  $P$  if no point in  $P$  dominates it.

Design an efficient algorithm to find all the maximal points of a given set  $P$ , where  $P$  contains  $n$  points.

**NAME:** \_\_\_\_\_

*this page left blank for work*

NAME: \_\_\_\_\_

**9: (14 points) Scheduling to Minimize Total Time**

Suppose you are given a set  $S = \{ a_1, a_2, \dots, a_n \}$  of tasks, where task  $a_i$  requires  $p_i$  units of processing time to complete, once it has started. You have one computer on which to run these tasks, and the computer can run only one task at a time. Let  $c_i$  be the completion time of task  $a_i$ , that is, the time at which task  $a_i$  completes processing. Assume each task starts immediately after the previous one has completed. Your goal is to minimize the total completion time of all tasks, that is, to minimize  $\sum_{i=1}^n c_i$ . For example, suppose there are two tasks,  $a_1$  and  $a_2$ , with  $p_1 = 3$  and  $p_2 = 5$ , and consider the schedule in which  $a_2$  runs first, followed by  $a_1$ . Then  $c_2 = 5$ ,  $c_1 = 8$ , and the total completion time is  $5 + 8 = 13$ .

Give an algorithm that schedules the tasks so as to minimize the total time. Each task must run non-preemptively, that is, once task  $a_i$  is started, it must run continuously for  $p_i$  units of time. Prove that your algorithm minimizes the total time, and state the running time of your algorithm.

**NAME:** \_\_\_\_\_

*this page left blank for work*