

This exam consists of 6 questions on a total of 10 pages, worth a total of 70 points. There is an extra (empty) page included at the end. Please provide the answer at the space right under the question. You may also use the back of this page or the last page for additional space, but please mark on the question page itself if you are using this as part of an answer. **You have 2 hours and 15 minutes to complete this exam.**

This exam is closed-book and closed-notes. When you are asked to design and analyze an algorithm, you must present a full description of the algorithm and an analysis of its running time. The algorithm's running time is required to be polynomial in the input size. **You may reduce problems to ones presented in textbooks, lectures, or homework**, in which case you need not explain the details of the algorithm, and can simply state its running time. However, you still must provide the full protocol to reduce to the existing problem, as well as a correct running time analysis of the combined process of reducing the new problem to the existing one and executing the algorithm for the reduced problem. On this exam, to save time, **please only prove what we ask for in each problem.**

## Assignment Project Exam Help

**Important:** Follow the directions to each question carefully. If you are printing the exam, place answers in designated spaces. If you are writing the exam on separate paper, clearly label which of your work corresponds to each question.

STATEMENT:

- I am aware that my score on this exam may increase or lower my final letter grade.
- I took this exam without accessing any course materials or notes.
- I understand that academic integrity violations on this exam result in an F in the course.

SIGNATURE: \_\_\_\_\_

(If you did not print the exam, please state at the top of your first page that you read all instructions, and sign your statement.)

Net ID: \_\_\_\_\_

This page can be used as extra space. Please mark on the question page itself if you are using this page for part of an answer.

**Assignment Project Exam Help**

**<https://tutorcs.com>**

**WeChat: cstutorcs**

**Part I: Short Answer Questions** (*31 points*)

1. ( $4 \times 4$  points) Select whether the following statements are true or false, and **give a brief explanation (at most two sentences)**.

(a) ☐ **True** / ☐ **False** : The SUBSET SUM instance with  $w_1, \dots, w_n$  and  $W$  is solvable in polynomial time (in  $n$ ) if  $W \leq n$ .

(b) Given a SAT formula with  $n$  variables and  $m$  clauses, AT-LEAST-TWO-SAT asks whether there is a truth assignment that satisfies at least 2 of the clauses. Assuming P is not equal to NP,

☐ **True** / ☐ **False** : AT-LEAST-TWO-SAT is NP-hard because SAT  $\leq$  AT-LEAST-TWO-SAT.

(c) Consider a divide-and-conquer algorithm that solves a problem of size  $n$  by solving  $q \geq 2$  subproblems of size  $n/2$ , and takes  $O(n)$  time to combine the results from the subproblems.

☐ **True** / ☐ **False** : The running time of this algorithm is linear in  $n$ .

- (d) You are editing some old code at an internship, and notice a line of code that looks odd. You wonder if this instruction will ever be executed, or if it can be deleted.

You realize that you can make this into a question about Turing Machines: given a Turing Machine  $M$  and a specific state  $q$  of  $M$ , is there any input  $x$  on which  $M$  will at some point be in state  $q$ ?

☐ **True** / ☐ **False** : This problem is undecidable.

2. (2+4 points) Consider a bipartite graph  $G = (V, E)$  on  $2n$  nodes, where  $V = \{1, 2, \dots, 2n\}$  and the two sides of the partition are  $\{1, 2, \dots, n\}$  and  $\{n+1, n+2, \dots, 2n\}$ . Consider the following greedy algorithm for finding a maximum size matching in  $G$ :

```

 $M \leftarrow \emptyset.$ 
for  $i = 1$  to  $n$ :
    if node  $i$  has any unmatched neighbors:
        Let  $j$  be the unmatched neighbor of  $i$  with the smallest index.
        Add  $(i, j)$  to  $M$ .
    end if
end for
return  $M$ .

```

While this algorithm is not guaranteed to find a maximum matching, it has an approximation guarantee: there is some  $\alpha$  such that the matching returned by the algorithm will have size at least  $1/\alpha$  times the size of the maximum matching.

- (a) Give the best possible value for  $\alpha$ :  $\alpha = \underline{\hspace{1cm}}$ .
- (b) Show that your answer to (a) is *tight*. That is, give an instance for which the greedy algorithm returns a matching that is exactly a factor  $1/\alpha$  smaller than the maximum matching.

3. ( $3 \times 3$  points) For each of the following binary languages, select the appropriate classification. Here, “RE” abbreviates Recursively Enumerable.

Give a brief (at most two sentence) explanation of your answers. (*Hint: you should use each classification exactly once.*)

$L_1 = \{M : M \text{ is a binary string describing a Turing Machine that accepts the empty string } \}$ .

☐ Recursive                      ☐ RE, Not Recursive                      ☐ Not RE

Explanation:

## Assignment Project Exam Help

$L_2 = \{M : M \text{ is a binary string describing a Turing Machine that rejects its own description, } M, \text{ when passed in as input } \}$ .

☐ Recursive                      ☐ RE, Not Recursive                      ☐ Not RE

Explanation:

WeChat: cstutorcs

$L_3 = \{M : M \text{ is a binary string, and its length is prime } \}$ .

☐ Recursive                      ☐ RE, Not Recursive                      ☐ Not RE

Explanation:

**Part II: Long Answer Questions (39 points)**

4. (2+14 points) A electronic toy company must include resistors between certain components of a toy's circuitry. More specifically, if they need to ensure a resistance  $R$  between two components, they will need to include individual resistors whose resistance values sum to  $R$ <sup>1</sup>. Each resistor, regardless of its resistance, has the same cost, so the company wishes to minimize the number of resistors used. That is, given a set of possible resistor types  $r_1 < \dots < r_k$ , the company must select the number  $n_i$  of units of resistor  $r_i$  to buy for each  $1 \leq i \leq k$ , such that  $R = \sum_{i=1}^k (n_i \cdot r_i)$  and  $\sum_{i=1}^k n_i$  is minimized.

Currently, the toy company works with a supplier that sells  $1\Omega$ ,  $4\Omega$  and  $16\Omega$  resistors. To determine which resistors to buy, the company utilizes a greedy procedure that considers the resistors in decreasing order resistance  $r_i$ , and takes as many of each as possible without going over the target resistance  $R$ :

```

for  $i = k$  to 1:
     $n_i \leftarrow \lfloor \frac{R}{r_i} \rfloor$ 
     $R \leftarrow R - n_i \cdot r_i$ 
end for

```

This procedure returns the optimal selection for this set of resistors.

- (a) The supplier introduces a new  $9\Omega$  resistor to their catalog. Argue that the greedy procedure is no longer always optimal.

- (b) Due to supply chain issues, the toy company would like the flexibility to switch between suppliers. However, each supplier offers a different set of resistor types. Design an algorithm that takes in a set of resistor types  $r_1 < \dots < r_k$  and a target resistance  $R$ , and returns the optimal selection  $(n_1, \dots, n_k)$  as described above. If the target resistance cannot be achieved (for example, if each  $r_i$  is even, but  $R$  is odd), this should be reported.

Assume all inputs are positive integers.

Give a pseudo-polynomial-time algorithm to solve this problem. Your algorithm should return the number of each resistor type  $(n_1, \dots, n_k)$  or report that no solution can achieve the target resistance. You should **give the algorithm and analyze the running time**, but you do **not** have to give a proof of correctness. Pseudocode is acceptable, as long as you explain in English the intended meaning of your variables, and you **explain the reasoning behind your algorithm**.

(Write your answer on the following page.)

---

<sup>1</sup>Here, we are only concerned with wiring resistors in series, so the total resistance equals the sum of the resistances.

Net ID: \_\_\_\_\_

(Write your answer to Problem 5 here.)

Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs

5. (8+5 points) A company wishes to treat its employees to donuts. An assorted box from Edmond's donuts contains one donut of each of their 12 flavors. From each of their  $n$  employees, the company collects a non-empty list  $D_i \subseteq \{1, \dots, 12\}$  of the donut flavors that employee  $i$  likes. They must determine how many assorted boxes of donuts they should order so that every employee gets a donut that they like.

In the following two subparts, you are asked to design two algorithms. You must fully **explain how your algorithms work** (for example, explain the meanings of all variables other than loop counters) and **analyze their running times**, but you do **not** need to provide proofs of correctness.

- (a) Design a polynomial time algorithm that takes as input the  $n$  flavor lists  $D_1, \dots, D_n$  and an integer  $b$ . The algorithm should return whether ordering  $b$  boxes is sufficient to ensure everyone gets a donut that they like.

## Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs

- (b) Design a polynomial time algorithm that takes as input the  $n$  flavor lists  $D_1, \dots, D_n$  and returns the minimum number of boxes that must be purchased to ensure everyone gets a donut that they like. *Regardless of your progress, you may assume that you have a correct and sufficiently fast implementation of the algorithm from part (a).* **For full credit, your algorithm's runtime should match our solution's.**



6. (10 points) An elementary school teacher has a class of  $n$  students. For each student  $s$ , they have a list of that student's friends. Assume that these friendship lists are symmetric, so if  $t$  is on  $s$ 's friend list, then  $s$  is also on  $t$ 's list. We can model this friendship information as a graph  $G$  on  $n$  nodes, where (undirected) edges represent mutual friendships.

Suppose that on a particular day, some subset  $A$  of the students are absent. The teacher has a worksheet that they would like to pass along to the absent students. The DELIVER PAPERS problem **takes as input**  $G$ ,  $A$ , **and an integer**  $k$  and asks whether there is a subset  $P$  of students such that:

- $|P| \leq k$ .
- Each student in  $P$  is in school (i.e. not absent).
- Each student in  $A$  is friends with at least one student in  $P$ .

DELIVERPAPERS is in NP (you do not have to prove this).

Prove that DELIVERPAPERS is NP-hard. That is, argue that  $Y \leq_p$  DELIVERPAPERS for some NP-complete problem  $Y$ .

## Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs

Net ID: \_\_\_\_\_

This page can be used as extra space. Please mark on the question page itself if you are using this page for part of an answer.

**Assignment Project Exam Help**

**<https://tutorcs.com>**

**WeChat: cstutorcs**