CS 5300 Advanced Algorithm Design and Analysis
Final Exam
Posted on Canvas: Class Time 12/7/2023
Due: 2:50PM, 12/14/2023
(Submission to Canvas)

Name : ______________________

Last 4 digits of your Student ID #: ______________________

Read these instructions before proceeding.

- Your answers should be typed on 8.5 x 11 inches white paper, double-spaced. Use this page as your cover page. Write your name and the last 4 digits of your Student ID number at the top right. After grading, your name and the last 4 digits of your Student ID number will be matched so that your grade can be correctly recorded.

- Answer to each question is **limited to be one to two pages**.

- You need to turn in your answers for this Final-Exam on canvas.cpp.edu no later than **2:50PM, 12/14/2023.**

- No questions will be answered for this exam by emails or during my office hours.

- Answer each question the best you can. Partial credit will be awarded for reasonable efforts. If a question contains an ambiguity or a misprint, then say so in your answer, providing the answer to a reasonable interpretation of the question; give your assumptions.

- Open book. Open notes. Open to any source. If your answer is based on a particular article or a source, you need to understand it completely and write the answer on your own. Also remember to quote it properly and stick to a style, e.g. IEEE or MLA.

- There are six questions. The total is 100 points.

- Good luck!

Q#1 (16)	Q#2 (16)	Q#3 (15)	Q#4 (22)	Q#5 (16)	Q#6 (15)	Total (out of 100)

1. (16 pts)
(Length of your answer: 1-2 pages)

Question:
Decide ***True*** or ***False*** for the following questions. In each question, justify your answer.

(a) All NP-complete problems are NP-hard.

(b) All NP-hard problems are NP-complete.

(c) If a problem A is NP-complete, proving that A is reducible to B, in polynomial time, is sufficient to show that B is NP-complete.

(d) If a problem C is NP-hard, proving that C is reducible to D, in polynomial time, is sufficient to show that D is NP-hard.

2. (16 pts)
(Length of your answer: 1-2 pages)

THREE2ONE-PARTITION Problem :
Instance: A finite set of positive integers $Z = \{z_1, z_2, \ldots, z_n\}$.
Question: Is there a subset Z' of Z such that
Sum of all numbers in Z' = 3 × Sum of all numbers in $Z - Z'$

SUM-OF-SUBSETS Problem
Instance: A finite set $A = \{a_1, a_2, \ldots, a_m\}$ and M.
Question: Is there a subset A' in A s.t. $\sum_{a_i \text{ in } A'} a_i = M$?

Given that THREE2ONE-PARTITION Problem is NP-Complete, **prove that the SUM-OF-SUBSETS Problem is NP-Complete** by reducing THREE2ONE-PARTITION Problem to it.

(a) Give a nondeterministic polynomial time algorithm for the SUM-OF-SUBSETS Problem. (Use *Guess* statements in your solution, e.g. *Guess*({0,1}) returns 0 or 1)

(b) Define the transformation from the THREE2ONE-PARTITION Problem to the SUM-OF-SUBSETS Problem and give the if-and-only-if proof.

3. (15 pts)
(Length of your answer: 1-2 pages)

Question:
Give the most interesting Final Exam question that you can think of from what you have learned in this class. State this question precisely and give a sample solution to this question. Explain why this is the most interesting.

4. (22 pts)
(Length of your answer: 1-2 pages)

<u>Question</u>:
Optimization PS(k) Problem: Given a set of n program and k storage devices (disks). Let s_i be the length (amount of storage) needed to store the i^{th} program. Let L be the storage capacity of each storage device. Determine the maximum number of these n programs that can be stored on k storage devices (without splitting a program over the storage devices).

Use the Approximation PS Algorithm (smallest-program-first) covered in the class for the PS(k) problem given above.

(a) (10 pts) Let the approximation PS algorithm returns a number C, and let C^* be the optimal (maximum) number of programs that can be stored on the k disks. Show that the above approximation PS algorithm gives the performance ratio of $C^* \le (C + k - 1)$.

(b) (6 pts) Consider $k = 4$ and $C^* \le (C + 3)$.
Give an example (set of programs with *lengths* and a value L) that achieves the performance ratio of $C^* = (C + 3)$ for the 4-disk problem of PS(4), and show that it is true.
Hints: Example for PS(2): 4 programs with lengths of 1,1,2,2 and L=3. Example for PS(3): 9 programs with lengths of 2,2,2,3,3,3,3,3,3 and L=8.)

(c) (6 pts) Consider $k = 5$ and $C^* \le (C + 4)$.
Give an example (set of programs with *lengths* and a value L) that achieves the performance ratio of $C^* = (C + 4)$ for the 5-disk problem of PS(5), and show that it is true.
Hints: Example for PS(2): 4 programs with lengths of 1,1,2,2 and L=3. Example for PS(3): 9 programs with lengths of 2,2,2,3,3,3,3,3,3 and L=8. Example for PS(4) you gave in part (b).

5. (16 pts)
(Length of your answer: 1-2 pages)

<u>Question</u>:
Look around your daily life, and find out two interesting real-life NP-hard problems. Turn in a picture (yes, take a picture from you) of each of these two problems and state the two problems clearly. Also explain why you think they are NP-hard.

6. (15 pts)
(Length of your answer: 1-2 pages)

<u>Question</u>:
Search and find three songs/movies/paintings/artworks that are related to Algorithms or NP-hardness.

Put down a title and an URL (link) for each of your three answers, and summarize how each of them is related to Algorithms or NP-hardness.