MA/CSSE 473 HW 15 textbook problems and hints August 4, 2011

Problem #1 (5) 11.1.1 (lower bound for alternating disk algorithm)

1. Prove that any algorithm solving the alternating-disk puzzle (Problem 11 in Exercises 3.1) must make at least $n(n+1)/2$ moves to solve it. Is this lower bound tight?

Author's hint:

1. Is it possible to solve the puzzle by making fewer moves than the brute-force algorithm? Why?

Problem #2 (5) 11.1.4 (fake coin minimum number of guesses)

4. Consider the problem of identifying a lighter fake coin among n identical-looking coins with the help of a balance scale. Can we use the same information-theoretic argument as the one in the text for the number of questions in the guessing game to conclude that any algorithm for identifying the fake will need at least $\lceil \log_2 n \rceil$ weighings in the worst case?

Author's hint:

4. Reviewing Section 5.5, where the fake-coin problem was introduced, should help in answering the question.

Problem #3 (12) 11.1.10 (matrix multiplication and squaring) (6, 6)

10. a. Can we use this section's formulas that indicate the complexity equivalence of multiplication and squaring of integers to show the complexity equivalence of multiplication and squaring of square matrices?

    b. Show that multiplication of two matrices of order n can be reduced to squaring a matrix of order $2n$.

Author's hint:

10. a. Check whether the formulas hold for two arbitrary square matrices.

    b. Use a formula similar to the one showing that multiplication of arbitrary square matrices can be reduced to multiplication of symmetric matrices.

Problem #4 (9) 11.2.10ab [11.2.8ab] (advanced fake-coin problem) (4, 5)

8. *Advanced fake-coin problem* There are $n \geq 3$ coins identical in appearance; either all are genuine or exactly one of them is fake. It is unknown whether the fake coin is lighter or heavier than the genuine one. You have

a balance scale with which you can compare any two sets of coins. That is, by tipping to the left, to the right, or staying even, the balance scale will tell whether the sets weigh the same or which of the sets is heavier than the other, but not by how much. The problem is to find whether all the coins are genuine and, if not, to find the fake coin and establish whether it is lighter or heavier then the genuine ones.

a. Prove that any algorithm for this problem must make at least $\lceil \log_3(2n+1) \rceil$ weighings in the worst case.

b. Draw a decision tree for an algorithm that solves the problem for $n = 3$ coins in two weighings.

Author's hint

8. a. How many outcomes does this problem have?

   b. Draw a ternary decision tree that solves the problem.

:

Problem #5 (5) 11.3.1 (Chess decidable?) Explain your answer.

1. A game of chess can be posed as the following decision problem: given a legal positioning of chess pieces and information about which side is to move, determine whether that side can win. Is this decision problem decidable?

.

Author's hint:

1. Check the definition of a decidable decision problem.

Problem #6 (8) 11.3.2 (tractable?) Explain your answer.

2. A certain problem can be solved by an algorithm whose running time is in $O(n^{\log_2 n})$. Which of the following assertions is true?

   a. The problem is tractable.

   b. The problem is intractable.

   c. None of the above.

Author's hint:

2. First, determine whether $n^{\log_2 n}$ is a polynomial function. Then read carefully the definitions of tractable and intractable problems.

Problem #7 (5) 11.3.6 (brute force composite number)

6. Consider the following brute-force algorithm for solving the composite number problem: Check successive integers from 2 to $\lfloor n/2 \rfloor$ as possible

divisors of n. If one of them divides n evenly, return yes (i.e., the number is composite), if none of them does, return no. Why does this algorithm not put the problem in class P?

Author's hint:

6. What is a proper measure of an input's size for this problem?

Problem #8 (5) 11.3.7a (polynomial –time check of knapsack solution)

7. State the decision version for each of the following problems and outline a polynomial-time algorithm that verifies whether or not a proposed solution solves the problem. (You may assume that a proposed solution represents a legitimate input to your verification algorithm.)

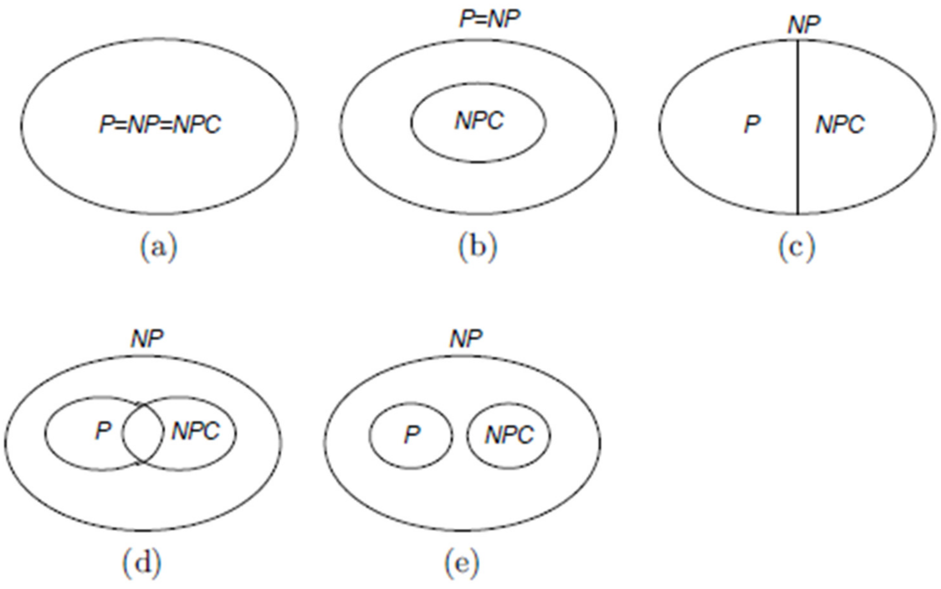
   a. knapsack problem

Author's hint:

7. See the formulation of the decision version of graph coloring and the verification algorithm for the Hamiltonian circuit problem given in Section 11.3.

Problem #9 (5) 11.3.11 [11.3.10] (Venn diagrams)

10. ▷ Which of the following diagrams do not contradict the current state of our knowledge about the complexity classes P, NP, and NPC (NP-complete problems)?



Author's hint:

10. Just two of them do not contradict the current state of our knowledge about the complexity classes.

Problem #10 (10) 11.3.12 [11.3.11] (King Arthur problem) Optional, extra-credit problem

11. King Arthur expects 150 knights for an annual dinner at Camelot. Unfortunately, some of the knights quarrel with each other, and Arthur knows who quarrels with whom. Arthur wants to seat his guests around a table so that no two quarreling knights sit next to each other.

    a. Which standard problem can be used to model King Arthur's task?

    b. As a research project, find a proof that Arthur's problem has a solution if each knight does not quarrel with at least 75 other knights.

5. Determine the time-efficiency class of the stable-marriage algorithm

    (a) in the worst case.

    (b) in the best case.

Author's hint:

11. The problem you need is mentioned explicitly in Section 11.3.