# CMPS/EECE 500
# Homework #5

1. Compare and contrast class P, NP.

2. Does rational knapsack problem belongs to NP-class? Explain your answer with details.

3. Show all the details in transforming a sorting algorithm, say merge sort, into a decision problem.

4. Find all assignments satisfying the following Boolean formulas

$p \vee q \vee \neg r$
$r \vee \neg q$
$r \vee \neg p$
$\neg q \vee \neg r$

5. Does the following set of clause satisfiable? If the answer is yes, provide the satisfying interpretation.

$$\left[\begin{array}{l} p \vee q \vee \neg s \\ r \vee \neg p \\ s \vee q \vee \neg p \\ s \vee \neg s \\ p \vee \neg q \vee \neg s \\ s \vee \neg q \end{array}\right.$$

6. John, a brilliant CACS Ph.D. student, has developed an algorithm to solve an NP-complete problem. He has demonstrated it by implementing his algorithm on his PC and running it with few data sets each is in the order of few hundreds. The running time of his algorithm on his data sets are consistently in the order of few minutes. His advisor was stunned and they are planning to have a press conference to announce the "so called breakthrough." What advice, if any, you may provide to the student and his advisor before they go public with the results?

7. Compare and contrast under constrained, critically constrained and over constrained problems.

8. Show that 3SAT is NP. Assuming SAT is NP-complete, show that 3SAT is NP-complete. (Show all the relevant steps)

1. Compare and contrast class P, NP

solution

P is the class of decision problems that can be solvied in polynomial time

NP is the class of decision problems that can be verified in polynomial time

All problems that are in P class are also in NP class because if it takes polynomial times to find a solution to a decision problem then it can be also verified in polynomial time i.e. ~~P is a~~

P is a subset of NP

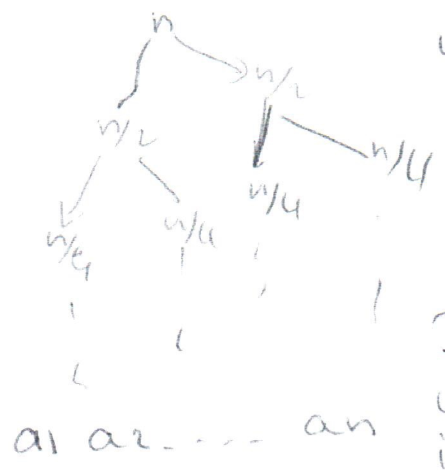
Vice versa cannot be true always as there may be set of problems that take polynomial time to verify an answer for, but there exits no method to solve these problems in polynomial time

2] ~~solution~~. Does rational Knapsack problem belongs to NP-class? Explain your answer with details.

solution: Rational Knapsack problem belongs to NP-class the greedy strategy is used to compute rational Knapsack problem and find the profitable way to select items. We can solve this problem in O(n logn) time (which is polynomial time). therefore, a rational Knapsack problem is in P class as solution exists in polynomial time. By definition N class is a subset of NP-class, this makes the problem a member of NP-class

] solutions A decision problem is a "yes" or "no" question

Consider an array A of 'n' elements to be sorted. Merge sort is lgorithm based on divide & conquer strategy. It first break the array A[1] - - - A[n] into halves ~~on~~ each step until leaf ~~&~~ nodes (elements) are obtained.




when we start combining individual elements with respect to their values, we have to make decision to ~~make~~ find the smaller elements among them. therefore we have to make many decision i.e. is $a_1 > a_2 > a_3$ ...

is $a_i > a_{i+1}$

Decisions are done from leaves to root node until the sorted array is obtained and finally is the array sorted or not?

4. Find all assignments satisfying the following Boolean formulas

$p \vee q \vee \neg r$
$r \vee \neg q$
$r \vee \neg p$
$\neg q \vee \neg r$

the solutions

| p | q | r |
|---|---|---|
| 0 | 0 | 0 |
| 1 | 0 | 1 |

| p | q | r | $p \vee q \vee \neg r$ | $r \vee \neg q$ | $r \vee \neg p$ | $\neg q \vee \neg r$ |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | T | T | T | T |
| 0 | 0 | 1 | F | T | T | T |
| 0 | 1 | 0 | T | F | T | T |
| 0 | 1 | 1 | T | T | T | F |
| 1 | 0 | 0 | T | T | F | T |
| 1 | 0 | 1 | T | T | T | T |
| 1 | 1 | 0 | T | F | F | T |
| 1 | 1 | 1 | T | T | T | F |

5. Does the following set of clause satisfiable ? if the answer is yes, provide the satisfying interpretation

$p \vee q \vee \neg s$
$r \vee \neg p$
$s \vee q \vee \neg p$
$s \vee \neg s$
$p \vee \neg q \vee \neg s$
$s \vee \neg q$

| p | q | r | s | $p \vee q \vee \neg s$ | $r \vee \neg p$ | $s \vee q \vee \neg$ |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | | | |
| 0 | 0 | 0 | 1 | | | |
| 0 | 0 | 1 | 0 | | | |
| 0 | 0 | 1 | 1 | | | |
| 0 | 1 | 0 | 0 | | | |
| 0 | | | | | | |
| 0 | 1 | 0 | 1 | | | |
| 1 | 1 | | | | | |
| 1 | 1 | 1 | 0 | | | |
| 1 | 0 | 1 | 1 | | | |
| 1 | 0 | 0 | 0 | | | |
| 1 | 0 | 0 | 1 | | | |
| 1 | 1 | 1 | | | | |
| 1 | 1 | 1 | 0 | | | |
| 1 | | 0 | | | | |

6. Solution: First they must check the instances of the problem for the level of constraints. the instances may be either over constrained or under constrained and such cases any algorithm or heuristics for a given NP-complete problem will find solution in reasonable time comparable to polynomial time.
if the data instances fall into critically constrained and their algorithm solves the prob, then it is worth to announce the results
Finding non randomized ~~problem~~ deterministic algorithm to solve NP-complete prob with critically constrained instances with
[illegible] is high unlikely [illegible] [illegible] for
Finding non deterministic algorithm to solve [illegible] NP-complete [illegible] prob with critically constrained instances is highly unlikely.

---

7. Solution: compare & contrast under constrained, critically constrained and over constrained problem

under-constrained: in the context of satisfiability problem, if we have large number of variables (propositions) and few number of clauses any truth assignment has a very high probability of satisfying the set of clauses.
Critically constrained: these problems are hard to solve if the mix of the variables and the number of clauses are [illegible] [illegible] very large number of truth assignment must be tried to reach a satisfying solution. In this region the probability of a given assignment satisfying the set of clauses is 0.5
Over constrained: As we increase the number of clauses, the scale will tip the scale will tip such that very few truth assignment can satisfy all the clauses. that is the ~~[illegible]~~ probability of satisfying set of clauses is getting very low, will approach 0. Hence we know immediately that the set of clauses are not [illegible] [illegible] [illegible] the assignment [illegible] [illegible]