CS760 Topics in Computational Complexity **Jul-Nov 2024**

Exercises

1. Prove that there is a one-one map from set of C++ programs to natural numbers. Prove that there is a one-one map from set of real numbers to set of decision problems (a decision problem is function from $\{0,1\}^*$ to $\{0,1\}$).

   It is known that there is no one-one map from set of real numbers to set of natural numbers. Combining this fact with the above two claims, argue that we cannot write a C++ program for every decision problem.

2. Argue that a given Turing machine with any alphabet size can be converted into one with alphabet size 2.

3. Argue that a two tape Turing machine can be converted into one tape Turing machine. How does the running time change for any given input?

4. Describe a one-tape Turing machine with $\{0,1\}$ alphabet that accepts all strings in $\{0,1\}^*$ which are palindrome.

5. (Arora Barak Exercise 1.9) Define a RAM Turing machine to be a Turing machine that has random access memory. We formalize this as follows: in addition to tapes, the machine has an infinite array A that is initialized to all blanks (one cell of the array can store one alphabet symbol of the TM). It accesses this array as follows. One of the machines tapes is designated as the address tape. Also the machine has two special alphabet symbols denoted by R and W and an additional state we denote by q_{access}. Whenever the machine enters q_{access}, if its address tape contains $\langle i \rangle R$ (where $\langle i \rangle$ denotes the binary representation of i) then the value $A[i]$ is written in the cell next to the R symbol. If its tape contains $\langle i \rangle W \sigma$ (where σ is some symbol in the machines alphabet) then $A[i]$ is set to the value σ.

   Argue that if a Boolean function f is computable within time $T(n)$ by a RAM TM, then it can be computed by a TM in time $O(T(n)^2)$.

   Do you think RAM TM is a good model for real-world computers?

6. Prove that for a system of linear equation, if there is a solution then there is solution whose size is polynomial in the input size.

7. Prove that for a system of linear equation, if there is no solution then there is certificate for this fact, and the certificate size is polynomial in the input size.

8. The integer factoring problem asks for computing all the prime factors of a given integer. What is a natural decision problem for this search problem? Prove that the decision problem is in NP $\cap$ coNP.

9. Consider a decision version of the discrete log problem, given numbers a, b, c and prime p, is there a number $1 \leq r \leq c$ such that $a^r \equiv b \pmod{p}$? Prove that this problem is in NP $\cap$ coNP. Note that there may be more than one values of r that satisfy the given equation.

10. Prove that P $\subseteq$ NP $\cap$ coNP.

11. Prove that if SAT is in coNP then NP = coNP.

12. Let us define the class exponential time as

$$\text{EXP} = \cup_{c \geq 1} \text{DTIME}(2^{n^c}).$$

    Similarly, we can define NEXP (non-deterministic exponential time). Show that if P = NP then EXP = NEXP.

13. If two languages L_1 and L_2 are in NP. Then is $L_1 \cup L_2$ in NP? Is $L_1 \cap L_2$ in NP?

14. Prove that undirected longest path problem is NP-hard.

15. Prove that max-cut problem is NP-hard (try a reduction from Independent set).

16. Prove that 3-coloring (for a given graph can vertices be colored with 3 colors) is NP-hard. Try a reduction from 3-SAT.

17. We are given a Boolean formula $\psi(x, y)$ in two sets of Boolean variables $x = \{x_1, x_2, \ldots, x_n\}$ and $y = \{y_1, y_2, \ldots, y_m\}$. For a given assignment to the variables, let #clause($\psi(\cdot)$) denote the number of clauses satisfied by the assignment. We want to find

$$\min_{x \in \{T,F\}^n} \max_{y \in \{T,F\}^m} \#\text{clause}(\psi(x, y)).$$

    Find an appropriate decision problem for this optimization problem. Can you say if the decision problem is in NP, or in coNP? Give an explanation.

18. Prove that if a graph has maximum independent set size k, then for any subset S of vertices, we have $|S| - 2|E[S]| \leq k$. Here $E[S]$ is the set of edges which connect vertices within S.

19. Give a polynomial time reduction from MAXCUT problem with parallel edges to MAXCUT problem without parallel edges.

20. Prove that if functions f and g are computable in logspace, then so is their composition $f \circ g$.

21. Prove that the following decision problem is in L. Given numbers a, b, c, decide whether $a \times b = c$.

22. If you have to write a definition of NL in terms of proof and verifier. What conditions will be put on the verifier?

23. Suppose there is an NP-hard problem A which has a polynomial time algorithm using oracles for problem B and problem C. Can you say that one of them have to be NP-hard?

24. Prove that 2-SAT is in NL.

25. Prove that the two definitions of logspace computable functions are equivalent. First definition has a write-only output tape. Second definition says that for any given i, the ith bit of the output can be computed in log-space.

26. Prove that if NP=coNP then PH=NP.

27. Prove that if $\Sigma_i^p = \Pi_i^p$ then PH= Σ_i^p.

28. Prove that the succinct tournament problem is in Π_2^p.

29. Prove that $\text{NP}^{\text{SAT}} \subseteq \Sigma_2^p$.

30. Prove that if EXP $\subseteq$ P/poly then EXP = Σ_2^p.

31. Prove that if multiplication was in AC^0 then so would be parity.

32. Prove that there is a language which has size $O(n^5)$ circuits, but not $O(n^2)$ size circuits.

33. A language L is called sparse if for every n, we have $|L \cap \{0,1\}^n| \leq O(n^c)$ for some constant c. Prove that if a sparse language is NP-complete then P=NP.

34. Prove that if SAT has a polynomial size circuit then we can also build a polynomial size circuit to find a satisfying assignment of a given CNF formula.

35. Prove that $\text{ACC}^0 \subseteq \text{TC}^0 \subseteq \text{NC}^1$.

36. Prove that $\mathrm{NC}^1 \subseteq \mathrm{L} \subseteq \mathrm{NL} \subseteq \mathrm{AC}^1$.

37. If NP is contained in P/log (polynomial time with logarithmic advice) then P = NP.

38. If SAT is in coNP/poly then polynomial hierarchy collapses to some level.

39. Suppose we have a randomized algorithm A that on input size n, uses R random bits and has success probability 3/4. That means, for every input x of size n, there are at least $3/4 \times 2^R$ choices of $r \in \{0,1\}^R$ such that $A(x,r)$ gives the correct answer.

    We want to prove that there exists a choice of $r_1, r_2, \ldots, r_k \in \{0,1\}^R$ such that for all inputs x of size n, majority of $A(x,r_1), A(x,r_2), \ldots, A(x,r_k)$ gives the correct answer. What would be your choice of k. You may need to use Chernoff bounds.

40. Use a counting argument to show that there exists a function $f\colon \{0,1\}^n \to \{0,1\}$ such that even to compute it correctly on $1/2 + 1/2^{n/10}$ fraction of inputs, you require an exponential size circuit.

41. Show that there exists a function $G\colon \{0,1\}^{10 \log n} \to \{0,1\}^n$ such that for all circuits C on n inputs and size $\leq n^2$, we have

$$\left| \Pr_{x \sim U_n}[C(x) = 1] - \Pr_{y \sim U_{10 \log n}}[C(G(y)) = 1] \right| \leq 1/10.$$

    Here U_n is the uniform distribution on length n strings.

42. Prove that BPP $\subseteq$ P/poly.

43. Prove that for any nonzero polynomial $f(x_1, x_2, \ldots, x_n)$ in n variables, if the degree of each variable is at most δ (called as individual degree bound), then

$$\Pr[f(a_1, a_2 ... a_n) \neq 0] \geq \left(1 - \frac{\delta}{|S|}\right)^n.$$

    Hint: induction on number of variables.

44. Design a polynomial item randomized algorithm for the following problem. Given a bipartite graph, where some edges are colored red, and given a number k, decide whether there is a perfect matching that contains exactly k red edges.

45. Prove that BPP $\subseteq \Sigma_2^p$.

46. Yao's theorem. Let Y be a distribution over $\{0,1\}^m$. Suppose that there exists $S > 10n$ and ϵ such that for every circuit C of size at most $2S$ and $i \in \{1, 2, \ldots, m\}$,

$$\Pr_{r \bar{Y}}[C(r_1, r_2, \ldots, r_{i-1}) = r_i] \leq \frac{1}{2} + \frac{\epsilon}{m}.$$

    Then prove that Y is (S, ϵ)-pseudorandom.

47. If there is a $2^{\ell/10}$-PRG, then there exists a function f computable in exponential time such that $H_{wrs}(f)(n) \geq 2^{3(n-1)/10}$.

48. We saw that if we have an exponential time computable function f with $H_{avg}(f) \geq 2^{2k/3}$, then BPP=P. Suppose we have slightly weaker lower bound. That is, we have an exponential time computable function f with $H_{avg}(f) \geq 2^{\sqrt{k}}$, then can we still say that every problem in BPP has a polynomial time deterministic algorithm? If not, then what is the best deterministic time complexity we can get?

49. Prove that if we have a polynomial time algorithm for 1/16-approximation of maximum independent set problem, then we will also have a polynomial time algorithm for 1/2-approximation of maximum independent set problem.