Name: Sanjoto, A netid: sanjoto2 uin:669383778

---

1. (3 points.) Consider a CNF formula F with m clauses. In F there are at most $\sqrt{m}$ clauses of size 3, and the rest are clauses of size 2. Deciding if such a formula is satisfiable is

(A) NP-Complete.

(B) None of the other answers are correct.

(C) Can not be solved in linear time, but can be done in polynomial time.

(D) Can be solved in linear time.

Correct answer: A.
Your answer: A.
3 out of **3** points received

---

2. (3 points.) Given a graph G, and vertices u and v, these two vertices are ***robustly connected***, if they remain connected, even if we remove any three vertices in G (except for u and v, naturally). Consider the problem of deciding if u and v are robustly connected.

(A) This problem is NP-Hard.

(B) This problem can be solved in polynomial time.

Correct answer: B.
Your answer: B.
3 out of **3** points received

3. (3 points.) Given two NFAs N_1 and N_2 with n_1 and n_2 states, respectively. Then there is a DFA M that accepts the language $L(N_1) \cap L(N_2)$.

(A) False.

(B) True, and the number of states of M is at most $n_1 n_2$.

(C) None of the other answers is correct.

(D) True, and the number of states of M is at most $O(n_1 + n_2)$.

(E) True, and the number of states of M is at most $2^{n_1} 2^{n_2}$.

Correct answer: E.
Your answer: A.
0 out of **3** points received

4. (2 points.) You are given a set $\mathcal{I} = \{I_1, I_2, \ldots, I_n\}$ of n weighted intervals on the real line. Consider the problem of computing the maximum weight set $\mathcal{C} \subseteq \mathcal{I}$, such that every pair of intervals of $\mathcal{C}$ intersect. This problem

(A) NP-Hard.

(B) Can be solved in polynomial time.

(C) NP-Complete.

(D) Can be solved in linear time by a greedy algorithm.

(E) Undecidable.

Correct answer: B.
Your answer: B.
2 out of **2** points received

5. (3 points.) Given k sorted arrays $A_1, A_2, \ldots, A_k$ with a total of n numbers stored in them (all numbers are distinct). Given a number x, one can compute the smallest number y, in these arrays, that is larger than x, in (faster is better)

(A) $O(n)$ time.

(B) $O(n \log n)$ time.

(C) $O(nk)$ time.

(D) $O(k \log n)$ time.

(E) $O(n^2)$ time.

Correct answer: D.
Your answer: D.
3 out of **3** points received

6. (2 points.) For a word $w = w_1 w_2 \ldots w_{2m}$, with $w_i \in \{0, 1\}^*$, let $w^{\mathrm{ODD}} = w_1 w_3 w_5 \ldots w_{2m-1}$. If a language L is a regular language, then the language $L^{\mathrm{ODD}} = \left\{ w^{\mathrm{ODD}} \;\middle|\; w \in L \right\}$ is regular.

(A) True.

(B) False.

Correct answer: A.
Your answer: A.
2 out of **2** points received

7. (2 points.) You are given a directed graph G with n vertices and m edges, a pair of vertices u, v, and a number t. Deciding if there is a closed walk, with at most t edges, that includes u and v can be done in

(A) $O(n \log n + m)$ time.

(B) None of the other answers is correct.

(C) $O(nm)$ time using Bellman-Ford.

(D) only exponential time since this problem is NP-Complete.

(E) $O(n + m)$ time.

Correct answer: E.
Your answer: E.
2 out of **2** points received

8. (3 points.) For text files S, T, let $\langle S, T \rangle$ denote the string that encodes the content of both S and T. Consider the language

$$L = \{\langle S, T \rangle \mid S, T \text{ are two java programs that stops on the same inputs}\}.$$

This language is

(A) Decidable.

(B) Context-free.

(C) Regular.

(D) Undecidable.

(E) None of the other answers.

Correct answer: D.
Your answer: D.
3 out of **3** points received

9. (3 points.) Consider the recurrence $f(n) = f\Big(\lfloor n/3)\rfloor\Big) + f\Big(\lfloor(n/2)\rfloor\Big) + O(n)$, where $f(n) = O(1)$ if $n < 10$. The solution to this recurrence is

(A) None of the above.

(B) $O(n)$.

(C) $O(n \log n)$.

(D) $O(1)$.

(E) $O(n^2)$.

Correct answer: B.
Your answer: B.
3 out of **3** points received

10. (3 points.) Let G be a directed graph with weights on the edges (the weights can be positive or negative). The graph G has n vertices and m edges. Computing the shortest **simple** path between two vertices in G can be done in:

(A) This is not defined if there are negative cycles in the graph. As such, it can not be computed.

(B) This is NP-Hard.

(C) This can be solved in $O(nm)$ time using Bellman-Ford.

(D) None of the above.

(E) This can be solved in $O(n \log n + m)$ time using Dijkstra.

Correct answer: B.
Your answer: E.
0 out of **3** points received

11. (2 points.) Consider a Turing machine (i.e., program) M that accepts an input $w \in \Sigma^*$ if and only if there is a CFG G such that $w \in L(G)$. Then the language of $L(M)$ is

(A) not well defined.

(B) finite.

(C) context-free.

(D) undecidable.

(E) Σ^*.

Correct answer: E.
Your answer: C.
0 out of **2** points received

12. (3 points.) Given a directed graph G with n vertices and m edges, consider the problem of deciding if there is a walk (the walk is allowed to repeat both vertices and edges) that visits at least half of the vertices of G.

(A) Can be solved in $O(n + m)$ time.

(B) NP-Complete.

(C) Can be solved in $O(nm)$ time, and no faster algorithm is possible.

(D) None of the above.

(E) NP-Hard.

Correct answer: A.
Your answer: C.
0 out of **3** points received

13. (5 points.) A string $s \in \{a, b\}^*$ is γ-*wild* if $|\#(a, s) - \#(b, s)| > \gamma$, where γ is a prespecified parameter, and $\#(a, s)$ is the number of appearances of a in s. For a string $w \in \{a, b\}^*$ and parameters γ and τ, breaking it into strings $s_1, s_2, \ldots, s_\ell$ is a ***(γ, τ)-breakup*** of w iff:

(I) For all i, $s_i \in \{a, b\}^*$.
(II) For all i, s_i is γ-wild.
(III) $w = s_1 s_2 \ldots s_\ell$ (that is, the concatenation of $s_1, s_2, \ldots, s_\ell$ is w),
(IV) $\ell \leq \tau$.

Given as input a string $w \in \{a, b\}^*$ of length m, and parameters γ and τ, an algorithm can decide if there is a (γ, τ)-breakup of w in (faster is better):

(A) $O(m^2 \tau)$ time.

(B) $O(m^2 \gamma)$ time.

(C) $O(m^3 \gamma)$ time.

(D) $O(m^3 \tau)$ time.

(E) $O(m^4)$.

Correct answer: A.
Your answer: B.
0 out of **5** points received

14. (3 points.) You are given an NFA N with n states (N might have ε-transitions), with the input alphabet being $\Sigma = \{0, 1\}$. Given a binary string $w \in \Sigma^*$ of length m, one can simulate N on a regular computer and decide if N accepts w. Which of the following is correct?

(A) None of the other answers is correct.

(B) This can be done in $O(n^m)$ time, and no faster algorithm is possible.

(C) This can be done in $O(2^n m)$ time, and no faster algorithm is possible.

(D) This problem can not be done in polynomial time, because it is undecidable.

(E) This can done in $O(n^2 m)$ time.

Correct answer: E.
Your answer: B.
0 out of **3** points received

15. (3 points.) Let $\mathcal{PC}$ be the class of all decision problems, for which there is a polynomial time certifier that works in polynomial time, and furthermore, for an input of length n, if it is a YES instance, then there is a certificate that is a binary string of length $n^{O(1)}$. We have that:

(A) NP $= \mathcal{PC}$.

(B) None of the other answers is correct.

(C) All the problems in $\mathcal{PC}$ can be solved in polynomial time.

(D) NP $\subsetneq \mathcal{PC}$.

(E) $\mathcal{PC}$ contains some NP-Complete problems, but not all of them.

Correct answer: A.
Your answer: C.
0 out of **3** points received

16. (3 points.) If a problem is NP-Hard, then it can also be undecidable. This statement is

(A) True if $P = NP$.

(B) None of the other answers.

(C) False if $P = NP$.

(D) True.

(E) False.

Correct answer: D.
Your answer: D.
3 out of **3** points received

17. (3 points.) Given an undirected graph G, with n vertices and m edges, consider the decision problem of determining if the vertices of G can be colored (legally) by 2 colors (i.e., no adjacent pair of vertices have the same color). This problem is:

(A) Can be solved in polynomial time.

(B) As hard as the independent set problem.

(C) NP-Complete.

(D) Solvable in $O(n + m)$ time.

(E) Undecidable.

Correct answer: D.
Your answer: A.
0 out of **3** points received

18. (1 point.) All problems in NP are solvable in exponential time. This statement is

(A) True.

(B) False.

Correct answer: A.
Your answer: B.
0 out of **1** point received

19. (2 points.) Consider the following decision problem: Given a directed graph G, and two vertices u, v in G, are u and v strongly connected in G?

This problem has a polynomial length certificate and polynomial time certifier. This claim is

(A) True.

(B) False.

Correct answer: A.
Your answer: A.
2 out of **2** points received

20. (3 points.) The number of undecidable languages is

(A) uncountable.

(B) undecidable.

(C) $2^{\mathbb{R}} = \aleph_2$.

(D) None of the other answers are correct.

(E) countable.

Correct answer: A.
Your answer: A.
3 out of **3** points received

21. (3 points.) Given an undirected graph G with n vertices and m edges, and a number k, deciding if G has a spanning tree with at most k leaves is

(A) Can be done in polynomial time.

(B) Can be done in $O(n+m)$ time.

(C) Can be done in $O(n \log n + m)$ time, and there is no faster algorithm.

(D) NP-Complete.

(E) Can be done in $O((n+m)\log n)$ time, and there is no faster algorithm.

Correct answer: D.
Your answer: E.
0 out of **3** points received

22. (2 points.) You are given an unsorted set X of n numbers. Deciding if there are two numbers x and y in X such that $x + y = 0$ can be solved in (faster is better):

(A) $O(n \log n)$ time.

(B) $O(n)$ time.

(C) $O(n^2)$ time.

(D) $O(n^{3/2})$ time.

(E) $O(n^2 \log n)$ time.

Correct answer: A.
Your answer: A.
2 out of **2** points received

23. (3 points.) For the language $L = \{a^n b^n \mid n \geq 0\}$, we have

(A) $F = \left\{a^i b^i \;\middle|\; i \geq 0\right\}$ is a fooling set for L.

(B) All of the sets suggested are fooling sets.

(C) $F = \{a^i \mid i \geq 0\}$ is a fooling set for L.

(D) None of the sets suggested are fooling sets.

(E) $F = \left\{a^i b^j \;\middle|\; i < j\right\}$ is a fooling set for L.

Correct answer: C.
Your answer: C.
3 out of **3** points received

24. (3 points.) For the following recurrence (evaluated from top to bottom in this order):

$$f(i,j,k) = \begin{cases} 1 & i < 0 \text{ or } j < 0 \text{ or } k < 0 \\ f(i-1,j,k)+1 & i > j \text{ or } i > k \\ f(i,j-1,k)+2 & j > k \\ f(i-1,j,k)+f(i,j-1,k)+f(i,j,k-1) & \text{otherwise.} \end{cases}$$

Assume that every arithmetic operation takes constant time (even if the numbers involved are large). Computing $f(n, \lfloor n/2 \rfloor, \lfloor n/4 \rfloor)$ can be done in (faster is better):

(A) $O(n^2)$ time, using dynamic programming.

(B) $O(n^3)$ time, using dynamic programming.

(C) $O(n)$ time, by recursion.

(D) $O(2^n)$.

(E) $O(n \log n)$ time.

Correct answer: B.
Your answer: B.
3 out of **3** points received

25. (3 points.) Give a CNF formula F with n variables, and m clauses, where every clause has exactly three literals (reminder: a literal is either a variable or its negation). Then, one can compute a satisfying assignment to F in:

(A) This is Satisfiability and it can not be solved in polynomial time unless $P = NP$.

(B) $O(n \log n + m)$ time.

(C) $O(n^2 + m^2)$ time.

(D) $O(2^n - 2^m)$ time.

(E) $O(n + m)$ time.

Correct answer: A.
Your answer: A.
3 out of **3** points received

26. (3 points.) Consider the problem of checking if a graph has a Hamiltonian path in it. This problem can be solved in

(A) It is NP-COMPLETE, so it can not be solved efficiently.

(B) None of the other answers are correct.

(C) Polynomial time.

(D) Maybe polynomial time – we do not know. Currently fastest algorithm known takes exponential time.

Correct answer: D.
Your answer: C.
0 out of **3** points received

27. (3 points.) Given an array $B[1 \,..\, n]$ with n real numbers (B is not sorted), consider the problem computing and printing out the smallest $\lfloor \sqrt{n} \rfloor$ numbers in B – the numbers should be output in sorted order. This can be done in

(A) $O(\sqrt{n} \log^2 n)$ time, and no faster algorithm is possible.

(B) $O(\sqrt{n} \log n)$ time, and no faster algorithm is possible.

(C) $O(n \log n)$ time, and no faster algorithm is possible.

(D) $O(n)$ time, and no faster algorithm is possible.

Correct answer: D.
Your answer: C.
0 out of **3** points received

28. (3 points.) You are given a directed graph G with n vertices, m edges, and positive weights on the vertices (but not on the edges). In addition, you are given two vertices u and v. The *weight* of a path π is the total weight of the vertices of π.

Consider the problem of computing the lightest (simple) path connecting a vertex u to a vertex v, that visits all the vertices of the graph. This problem is

(A) NP-Hard.

(B) Solvable in $O(n \log n + m)$ time.

(C) Polynomially equivalent to Eulerian cycle.

(D) Undecidable.

(E) Solvable in $O(n + m)$ time.

Correct answer: A.
Your answer: B.
0 out of **3** points received

29. (3 points.) You are given a directed graph G with n vertices and m edges. Consider the problem of deciding if this graph has k vertices $t_1, \ldots, t_k$, such that any vertex in G can reach any of these k vertices. This problem is

(A) NP-Complete by a reduction from Hamiltonian path/cycle to this problem.

(B) Doable in $O(n + m)$ time.

(C) None of other answers are correct.

(D) Doable in $O(n^k(n + m))$ time, and no faster algorithm is possible.

(E) NP-Complete by a reduction from this problem to Hamiltonian path/cycle.

Correct answer: B.
Your answer: A.
0 out of **3** points received

30. (3 points.) You are given a graph G with n vertices and m edges, and with weights on the edges. In addition, you are given a tree T. Verifying that T is an MST of G can be done in (faster is better):

(A) $O(n \log n + m)$ time.

(B) $O(nm)$ time algorithm, and no faster algorithm is possible.

(C) None of the other answers.

(D) $O(n)$ time.

(E) $O(\log n)$ time, after preprocessing the graph in $O(n)$ time.

Correct answer: A.
Your answer: C.
0 out of **3** points received

31. (2 points.) Consider an NFA N with m states defined over $\{0, 1\}^*$. There is an equivalent regular expression r (i.e., $L(r) = L(N)$), such that

(A) none of other answers are correct.

(B) r is of length at most $O(m)$.

(C) r is of length at most $O(m \log m)$.

(D) r is of length at most $f(m)$, where f is some function that is not specified in the other answers.

Correct answer: D.
Your answer: B.
0 out of **2** points received

32. (3 points.) You are given two algorithms A_Y and A_N. Both algorithms read an undirected graph G and a number k. If G has an independent set of size $\geq k$, then A_Y would stop (in polynomial time!) and output YES (if there is no such independent set then A_Y might run forever). Similarly, if G does not have an independent set of size $\geq k$, then the algorithm A_N would stop in polynomial time, and output NO (if there is such an independent set then A_N might run forever).

In such a scenario:

(A) This would imply that $P \neq NP$.

(B) One can in polynomial time output if G has a an independent set of size $\geq k$.

(C) Impossible since $P \neq NP$.

(D) This would imply that $P = NP$.

(E) At least two of the other answers are correct.

Correct answer: E.
Your answer: C.
0 out of **3** points received

33. (2 points.) Consider the language

$$L = \left\{ 1^i 2^j 3^k \mid i, j, k \geq 0, \text{ and } j \text{ is divisible by } p_1, p_2, \ldots, p_{100} \right\},$$

where p_j is the jth smallest prime number, for $j = 1, \ldots, 100$ (i.e., $p_1 = 2, p_3 = 3, \ldots, p_{100} = 541$). This language is

(A) Decidable.

(B) Finite.

(C) Undecidable.

(D) Context-free.

(E) Regular.

Correct answer: E.
Your answer: E.
2 out of **2** points received

34. (3 points.) Let $L_1, L_2 \subseteq \Sigma^*$ be context-free languages. Then the language $L_1 \cap L_2$ is always context-free.

(A) True only if the languages L_1 and L_2 are decidable, and no other answer is correct.

(B) False if the languages L_1 and L_2 are decidable, , and no other answer is correct.

(C) None of the other answers.

(D) True.

(E) False.

Correct answer: E.
Your answer: D.
0 out of **3** points received

35. (3 points.) Let $P_1, \ldots, P_{k+1}$ be $k+1$ decision problems. Consider a sequence of k polynomial reductions $R_1, \ldots, R_k$, where R_i works in quadratic time in its input size, and is a reduction from P_i to P_{i+1}. As such, there is a reduction from P_1 to P_{k+1} and its running time is

(A) $O(kn^2)$

(B) $O(k^2 n^2)$

(C) $O\big(n^{2k}\big)$

(D) $O\big(2^k n^2\big)$

(E) $O\Big(n^{2^k}\Big)$

Correct answer: E.
Your answer: C.
0 out of **3** points received

36. (3 points.) Let B be the problem of deciding if the shortest path in a graph between two given vertices is smaller than some parameter k (where the weights on the edges of the graph are positive).

Let C be the problem of deciding if a given CNF formula is satisfiable. Pick the correct answer out of the following:

(A) There is no relation between the two problems, and no reduction is possible.

(B) None of the other answers is correct.

(C) There is a polynomial time reduction from B to C.

(D) There is a polynomial time reduction from B to C, but only if $P = NP$.

(E) There is a polynomial time reduction from C to B, but only if $P \neq NP$..

Correct answer: C.
Your answer: D.
0 out of **3** points received

Summary of answers:

Question	Correct Answer	Your Answer	Points
1	A	A	3
2	B	B	3
3	E	A	0
4	B	B	2
5	D	D	3
6	A	A	2
7	E	E	2
8	D	D	3
9	B	B	3
10	B	E	0
11	E	C	0
12	A	C	0
13	A	B	0
14	E	B	0
15	A	C	0
16	D	D	3
17	D	A	0
18	A	B	0
19	A	A	2
20	A	A	3
21	D	E	0
22	A	A	2
23	C	C	3
24	B	B	3
25	A	A	3
26	D	C	0
27	D	C	0
28	A	B	0
29	B	A	0
30	A	C	0
31	D	B	0
32	E	C	0
33	E	E	2
34	E	D	0
35	E	C	0
36	C	D	0
Total			42