Final: Monday, December 18, 8-11am, 2017

A	B	C	D	E	F	G	H	J	K
9am	10am	11am	noon	1pm	1pm	2pm	2pm	3pm	3pm
Rucha	Rucha	Srihita	Shant	Abhishek	Xilin	Shalan	Phillip	Vishal	Phillip
101 Armory	101 Armory	101 Armory	151 Loomis	151 Loomis	151 Loomis	ECE 1002	ECE 1002	ECE 1002	ECE 1002

Name:	
NetID:	
Name on Gradescope:	

- ***Don't panic!***
- If you brought anything except your writing implements, your double-sided **handwritten** (in the original) 8½" × 11" cheat sheet, and your university ID, please put it away for the duration of the exam. In particular, please turn off and put away *all* medically unnecessary electronic devices.
  - Submit your cheat sheet together with your exam. An exam without your cheat sheet attached to it will not be graded.
  - If you are NOT using a cheat sheet, please indicate so in large friendly letters on this page.
- **Best answer.** Choose best possible choice if multiple options seems correct to you – for algorithms, faster is always better.
- Please ask for clarification if any question is unclear.
- **This exam lasts 170 minutes.**
- Fill your answers in the Scantron form using a pencil. We also recommend you circle/mark your answer in the exam booklet.
- Please return ***all*** paper with your answer booklet: your cheat sheet, and all scratch paper. We will **not** return the cheat sheet.
- Do not fill more than one answer on the Scantron form - such answers would not be graded. Also, fill your answer once you are sure of your answer – erasing an answer might make the form unscannable.
- ***Good luck!***

Before doing the exam...

- Fill your name and netid in the back of the Scantron form, and also on the top of this page.
- **Fill in the pattern shown on the right in the Scantron form.**

  This encodes which version of the exam you are taking, so that we can grade it.

	A	B	C	D	E
91	Ⓐ	Ⓑ	●	Ⓓ	Ⓔ
92	●	Ⓑ	Ⓒ	Ⓓ	Ⓔ
93	●	Ⓑ	Ⓒ	Ⓓ	Ⓔ
94	●	Ⓑ	Ⓒ	Ⓓ	Ⓔ
95	●	Ⓑ	Ⓒ	Ⓓ	Ⓔ
96	Ⓐ	Ⓑ	Ⓒ	Ⓓ	●

3

1. (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 $O((n+m)\log n)$ time, and there is no faster algorithm.
(C) Can be done in $O(n+m)$ time.
(D) Can be done in $O(n \log n + m)$ time, and there is no faster algorithm.
(E) NP-Complete.

---

2. (3 points) If a problem is NP-Complete, then it can also be undecidable. This statement is

(A) None of the other answers.
(B) False.
(C) True if $P = NP$.
(D) False if $P = NP$.
(E) True.

---

3. (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 instance of 2SAT formula is satisfiable. Pick the correct answer out of the following:

(A) There is a polynomial time reduction from C to B, but only if $P \neq NP$..
(B) There is a polynomial time reduction from B to C, but only if $P = NP$.
(C) There is a polynomial time reduction from B to C.
(D) None of the other answers is correct.
(E) There is no relation between the two problems, and no reduction is possible.

---

4. (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) finite.
(B) undecidable.
(C) Σ^*.
(D) context-free.
(E) not well defined.

5. (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) Polynomially equivalent to Eulerian cycle.
(B) NP-Hard.
(C) Solvable in $O(n \log n + m)$ time.
(D) Solvable in $O(n + m)$ time.
(E) Undecidable.

---

6. (3 points) You are given a DFA D with n states, and with the input alphabet being $\Sigma = \{0, 1\}$. Given a binary string $w \in \Sigma^*$ of length m, one can simulate D on a regular computer and decide if D accepts w. Which of the following is correct?

(A) None of the other answers is correct.
(B) This problem can not be done in polynomial time, because it is undecidable.
(C) This can be done in $O(2^n m)$ time, and no faster algorithm is possible.
(D) This can done in $O(n + m)$ time.
(E) This can be done in $O(n^m)$ time, and no faster algorithm is possible.

---

7. (3 points) You are given a directed graph G with n vertices and m edges. Consider the problem of deciding if there is a closed walk (the walk is allowed to repeat both vertices and edges) that visits all the vertices of G.

(A) This problem is NP-Complete.
(B) This problem is NP-Hard.
(C) This problem can be solved in $O(nm)$ time, and no faster algorithm is possible.
(D) This problem can be solved in $O(n + m)$ time.
(E) All of the other answers are correct.

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

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

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

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

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

---

9. (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) None of the above.

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

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

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

(E) This is NP-Hard.

---

10. (3 points) You are given a graph G with n vertices and $m = O(n)$ edges, and with weights on the edges. In addition, you are given the MST tree T of G.
Next, you are informed that the price of some edge e in the MST T had changed from its current cost (either increased or decreased), to a new cost α. Deciding if T is still the MST of the graph with the updated weights can be done in (faster is better):

(A) $O(n)$ time.

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

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

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

(E) $O(1)$ time.

11. (2 points) Consider a regular expression r that is of length m (i.e., writing r down requires m characters). Then, there is an equivalent NFA N with at most

(A) $2^{O(m)}$ states.

(B) 10 states.

(C) None of the other answers holds.

(D) $O(m)$ states.

(E) $m^{O(m^2)}$ states.

---

12. (2 points) For a word $w = w_1 w_2 \ldots w_m$, with $w_i \in \{0, 1\}^*$, let $w^Z = \overline{w_1} \ldots \overline{w_m}$, where $\overline{0} = 1$ and $\overline{1} = 0$. If a language L is a regular language, then the language $L^Z = \left\{ w^Z \,\middle|\, w \in L \right\}$ is regular.

(A) False

(B) True

---

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

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

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

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

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

---

14. (3 points) You are given a directed graph G with n vertices and m edges. Consider the problem of deciding if this graph has a vertex s, from which you can reach all the vertices of G. Solving this problem is

(A) NP-Hard by a reduction to Hamiltonian path/cycle.

(B) Doable in $O(n(n + m))$ time, and no faster algorithm exists.

(C) NP-Hard by a reduction from Hamiltonian path/cycle.

(D) NP-Hard by a reduction from 3SAT.

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

15. (3 points) You are given a graph G, and vertices u and v. Such a pair vertices is ***robustly connected***, if they remain connected, even if we remove any single vertex in G (except for u and v, naturally). Consider the problem of deciding if u and v are robustly connected.

(A) The problem is NP-Hard.

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

---

16. (3 points) For the language $L = \{a^n b^n \mid n \geq 0\}$, we have

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

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

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

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

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

---

17. (3 points) Let $\mathcal{NPL}$ 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 $O(\log n)$. We have that:

(A) $\mathcal{NPL}$ is an empty class of problems.

(B) $\mathcal{NPL}$ contains some NP-Complete problems.

(C) None of the other answers is correct.

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

(E) NP $\subseteq \mathcal{NPL}$.

---

18. (3 points) The number of decidable languages is

(A) undecidable.

(B) None of the other answers are correct.

(C) countable.

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

(E) uncountable.

---

19. (3 points) Given a DFA N and an NFA M with n and m states, respectively. Then there is a DFA M' that accepts the language $L(N) \setminus L(M)$.

(A) True, and the number of states of M is at most $n2^m$.
(B) True, and the number of states of M is at most $2^n 2^m$, and no other answer applies.
(C) False.
(D) True, and the number of states of M is at most nm.
(E) True, and the number of states of M is at most $(m+n)2^{(m+n)/2}$.

---

20. (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) Undecidable.
(B) Solvable in $O(n+m)$ time.
(C) NP-Complete.
(D) Can be solved in polynomial time.
(E) As hard as the independent set problem.

---

21. (2 points) You are given an unsorted set Y of m numbers. Deciding if there are two numbers x and y in Y such that $xy = 1$ can be solved in (faster is better):

(A) $O(m^{3/2})$ time.
(B) $O(m)$ time.
(C) $O(m \log m)$ time.
(D) $O(m^2)$ time.
(E) $O(m^2 \log m)$ time.

---

22. (3 points) Let $P_1, \ldots, P_{k+1}$ be $k+1$ decision problems in NP. 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, we have that

(A) $P_1, P_2, \ldots, P_k$ are polynomial time solvable.
(B) If P_1 is NP-Complete then P_{k+1} is NP-Complete (assuming k is a constant).
(C) If P_{k+1} is NP-Complete then P_1 is NP-Complete (assuming k is a constant).
(D) None of the other answers makes any sense. Also, this exam is stupid.
(E) $P_1, P_2, \ldots, P_k$ are NP-Complete.

23. (2 points) Consider the language

$$L = \left\{ 1^i \;\middle|\; i \geq 0, i \text{ is an integer, and } i \text{ is } \underline{\text{not}} \text{ divisible by } p_1, p_2, \ldots, p_{100} \right\},$$

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) Context-free.
(B) Finite.
(C) Decidable.
(D) Undecidable.
(E) Regular.

---

24. (3 points) For a text file T, let $\langle T \rangle$ denote the string that is the content of T. Consider the language

$$L = \{ \langle T \rangle \mid T \text{ is a java program that stops on some input} \}.$$

This language is

(A) Decidable.
(B) Context-free.
(C) Regular.
(D) Undecidable.
(E) None of the other answers.

---

25. (5 points) A binary string $s \in \{0, 1\}^*$ is ***k-balanced*** if $|\#(0, s) - \#(1, s)| \leq k$, where k is a prespecified parameter. For a string $w \in \{0, 1\}^*$ and parameters k and h, a split $u_1, u_2, \ldots, u_\ell$ is a ***(k, h)-valid split*** of w iff:

(I) For all i, $u_i \in \{0, 1\}^*$.
(II) For all i, u_i is k-balanced.
(III) $w = u_1 u_2 \ldots u_\ell$ (i.e., w is the concatenation of $u_1, u_2, \ldots, u_\ell$),
(IV) $\ell \leq h$.

Given as input a string $w \in \{0, 1\}^*$ of length m, and parameters k and h, an algorithm can decide if there is a (k, h)-valid split of w in (faster is better):

(A) $O(m^3 k)$ time.
(B) $O(m^3 h)$ time.
(C) $O(m^4)$.
(D) $O(m^2 k)$ time.
(E) $O(m^2 h)$ time.

26. (1 point) All problems in P are solvable in exponential time. This statement is

(A) True.

(B) False.

27. (3 points) Consider a CNF formula F with all clauses being of size 2, except for 10 clauses that are of size at most 7 (i.e., these clauses are made out of up to seven literals). Consider the problem of deciding if such a formula is satisfiable. We have:

(A) NP-Hard.

(B) Can be solved in linear time.

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

(D) None of the other answers are correct.

28. (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.

29. (3 points) Consider the recurrence $f(n) = f\Big(\lfloor n/2)\rfloor\Big) + f\Big(\lfloor (n/2)\rfloor\Big) + O(n \log n)$, where $f(n) = O(1)$ if $n < 10$. The solution to this recurrence is

(A) $O(n \log^2 n)$.

(B) None of the above.

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

(D) $O(n)$.

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

30. (3 points) Let $L_1 \subseteq \Sigma^*$ be a context-free language, and let $L_2 \subseteq \Sigma^*$ be regular. Then the language $L_1 \cap L_2$ is always context-free.

(A) False if the languages L_1 and L_2 are decidable.

(B) False.

(C) True if the languages L_1 and L_2 are decidable.

(D) True.

(E) None of the other answers.

31. (3 points) Given k sorted lists $L_1, L_2, \ldots, L_k$ with a total of n elements, one can compute the sorted list of all the elements in these lists in (faster is better):

(A) $O(nk)$ time.

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

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

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

(E) $O(n)$ time.

---

32. (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) Can be solved in polynomial time.

(B) NP-Hard.

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

(D) Undecidable.

(E) NP-Complete.

---

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

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

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

---

34. (3 points) Consider the problem of checking if a graph can be colored by four colors (i.e., no adjacent pair of vertices have the same color). It can be solved in

(A) Polynomial time.

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

(C) It is NP-Complete, so it can not be solved efficiently.

(D) None of the other answers are correct.

---

35. (2 points) Let G be a directed graph with n vertices and m edges. Deciding if two vertices u, v are in the same connected component of G can be done in

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

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

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

(D) None of the other answers is correct.

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

---

36. (3 points) Give a CNF formula F with n variables, and m clauses, where every clause has exactly two literals (reminder: a literal is either a variable or its negation). Then, one can compute a satisfying assignment to F in:

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

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

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

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

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