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.

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

7

1. (2 points) Consider the language $L = \{1^i \mid i \text{ is prime}\}$. This language is

(A) Decidable.

(B) Regular.

(C) Undecidable.

(D) Context-free.

(E) Finite.

---

2. (3 points) Let $\mathcal{PC}$ be the class of all decision problems, for which there is a certifier that works in polynomial time in the length of the input and the length of the certificate, 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 $2^{O(n)}$. We have that:

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

(B) None of the other answers is correct.

(C) NP $\subseteq \mathcal{PC}$.

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

---

3. (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) Σ^*.

(B) context-free.

(C) undecidable.

(D) finite.

(E) not well defined.

---

4. (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 $L^Z = \{w^Z \mid w \in L\}$ is regular.

(A) True

(B) False

---

5. (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) Undecidable.
(C) As hard as the independent set problem.
(D) NP-Complete.
(E) Solvable in $O(n+m)$ time.

---

6. (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) False.
(B) True.

---

7. (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) $O(1)$.
(B) None of the above.
(C) $O(n^2)$.
(D) $O(n \log n)$.
(E) $O(n)$.

---

8. (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) Can not be solved in linear time, but can be done in polynomial time.
(B) None of the other answers are correct.
(C) Can be solved in linear time.
(D) NP-Hard.

9. (3 points) For the language $L = \{0^n 1^n \mid n \geq 0\}$, we have

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

(B) $F = \{0^i 1^j \mid i < j\}$ is a fooling set for L.

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

(D) $F = \{0^i 1^i \mid i \geq 0\}$ is a fooling set for L.

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

---

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

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

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

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

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

---

11. (3 points) You are given an NFA N with n states (N might have ε-transitions), and 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 $w \notin L(N)$. Which of the following is correct?

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

(B) None of the other answers is correct.

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

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

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

---

12. (3 points) Given an array $B[1 \mathinner{..} n]$ 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) $O(\sqrt{n} \log n)$ time, and no faster algorithm is possible.

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

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

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

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

---

14. (2 points) You are given an unsorted set Z of n numbers. Deciding if there are two numbers x and y in Z such that $x = 1 - y$ can be solved in (faster is better):

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

---

15. (1 point) All problems in NP are solvable in exponential time. This statement is

(A) False.
(B) True.

---

16. (3 points) Let G be a DAG with weights on its edges (which can be either positive or negative). The DAG G has n vertices and m edges. Computing the shortest 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 can be solved in $O(n \log n + m)$ time using Dijkstra.
(C) This can be done in $O(n + m)$ time.
(D) This can be solved in $O(nm)$ time using Bellman-Ford.
(E) This is NP-Hard.

---

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

---

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

---

19. (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 $P = NP$.
(B) At least two of the other answers are correct.
(C) One can in polynomial time output if G has a an independent set of size $\geq k$.
(D) Impossible since $P \neq NP$.
(E) This would imply that $P \neq NP$.

20. (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.

---

21. (3 points) If a problem is NP-Complete, then it can also be undecidable. This statement is

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

---

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

---

23. (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, and a number t. The *weight* of a path π is the total weight of the vertices of π. Consider the problem of computing a (simple) path σ connecting a vertex u to a vertex v, such that the weight of σ is at least t. This problem is

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

---

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

---

25. (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 $O(n^k(n+m))$ time, and no faster algorithm is possible.
(C) NP-Complete by a reduction from this problem to Hamiltonian path/cycle.
(D) Doable in $O(n+m)$ time.
(E) None of other answers are correct.

---

26. (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.
(B) False if the languages L_1 and L_2 are decidable.
(C) True if the languages L_1 and L_2 are decidable.
(D) None of the other answers.
(E) True.

---

27. (3 points) The number of undecidable languages is

(A) undecidable.
(B) countable.
(C) None of the other answers are correct.
(D) uncountable.
(E) $2^{\mathbb{R}} = \aleph_2$.

28. (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.

---

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

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

(C) $O(n)$ time.

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

(E) $O(nk)$ time.

---

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

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

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

---

31. (5 points) Consider an array $A[1 \,..\, n]$ of n numbers. For an interval $I = [i \,..\, j]$ its *discrepancy* is the quantity $s(I) = \sum_{z=i}^{j} A[z]$. Such an interval I is k-*good*, if $s(I) \leq k$, where k is a prespecified parameter. Given A as above, and parameters k and u, a partition of $[1 \,..\, n]$ into ℓ intervals $I_1, \ldots, I_\ell$ is ***(k,u)-excellent*** iff:

(I) For all i, I_i is k-good.
(II) $[1 \,..\, n] = I_1 I_2 \cdots I_\ell$ (the concatenation of $I_1, I_2, \ldots, I_\ell$ is equal to $[1 \,..\, n]$),
(III) $\ell \leq u$.

An algorithm can decide if there is a (k,u)-excellent partition of A in (faster is better):

(A) $O(n^4)$.
(B) $O(n^2 u)$ time.
(C) $O(n^2 k)$ time.
(D) $O(n^3 k)$ time.
(E) $O(n^3 u)$ 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) NP-Hard.
(B) Undecidable.
(C) NP-Complete.
(D) Can be solved in linear time by a greedy algorithm.
(E) Can be solved in polynomial time.

---

33. (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) None of the other answers is correct.
(B) only exponential time since this problem is NP-Complete.
(C) $O(nm)$ time using Bellman-Ford.
(D) $O(n+m)$ time.
(E) $O(n \log n + m)$ time.

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

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

(C) None of the other answers is correct.

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

(E) False.

---

35. (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 the MST tree T.
Next, you are informed that the price of some edge e in the graph G had decreased from its current cost, 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) None of the other answers.

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

(C) $O(n)$ time.

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

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

---

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

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

(C) Polynomial time.

(D) None of the other answers are correct.