CS/ECE 374: Algorithms & Models of Computation, Fall 2017 **Instructor:** Sariel Har-Peled

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 Ⓐ ● Ⓒ Ⓓ Ⓔ

5

1. (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 $\lfloor \log^3 n \rfloor$ numbers in B that are larger than the median number in B – the numbers should be output in sorted order. This can be done in

(A) $O(n)$ time, and no faster algorithm is possible.
(B) $O(n \log n)$ time, and no faster algorithm is possible.
(C) $O(\log^4 n)$ time, and no faster algorithm is possible.
(D) $O(\log^3 n)$ time, and no faster algorithm is possible.

---

2. (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) $O(n \log n)$ time.
(B) $O(n \log k)$ time.
(C) $O(nk)$ time.
(D) $O(n^2)$ time.
(E) $O(n)$ time.

---

3. (3 points) Let $L \subseteq \Sigma^*$ be a context-free language. Then the complement language $\overline{L}$ is always context-free.

(A) True.
(B) None of the other answers.
(C) True if the language L is undecidable.
(D) False.
(E) True if the language L is decidable.

---

4. (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) $O(\log n)$ time, after preprocessing the graph in $O(n)$ time.
(B) $O(nm)$ time algorithm, and no faster algorithm is possible.
(C) $O(n)$ time.
(D) $O(1)$ time.
(E) $O(n \log n + m)$ time.

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

---

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

---

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

---

8. (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 $(m+n)2^{(m+n)/2}$.

9. (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.

---

10. (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) $O\left(2^k n^2\right)$
(B) $O\left(n^{2k}\right)$
(C) $O(k^2 n^2)$
(D) $O\left(n^{2^k}\right)$
(E) $O(kn^2)$

---

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

---

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

13. (3 points) The number of regular languages, over the alphabet $\{0, 1\}$, is

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

---

14. (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 $O(nm)$ time, and no faster algorithm is possible.
(B) None of the above.
(C) NP-Hard.
(D) NP-Complete.
(E) Can be solved in $O(n + m)$ time.

---

15. (3 points) For the following recurrence (evaluated from top to bottom in this order):

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

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

(A) $O(n \log n)$ time.
(B) $O(n)$ time, by recursion.
(C) $O(n^2)$ time, using dynamic programming.
(D) $O(2^n)$.
(E) $O(\log n)$.

---

16. (2 points) Consider the following decision problem: Given a directed graph G, and two vertices u, v in G, is there a path from u to v in G?
This problem has a polynomial length certificate and polynomial time certifier. This claim is

(A) True.
(B) False.

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 n 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) Solvable in $O(1)$ time.

---

18. (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) None of the other answers is correct.

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

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

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

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

---

19. (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 a value $x \in \mathbb{R}$, that maximizes the total weight of the intervals of $\mathcal{I}$ containing x. This problem:

(A) Undecidable.

(B) Can be done in polynomial time.

(C) NP-Complete.

(D) Can be done in linear time.

(E) NP-Hard.

20. (5 points) Consider an array $A[1 \mathinner{..} n]$ of n numbers. For an interval $I = [i \mathinner{..} j]$ 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 $[1 \mathinner{..} n]$ into ℓ intervals $I_1, \ldots, I_\ell$ is ***(k,u)-excellent*** iff:

(I) For all i, I_i is k-good.
(II) $[1 \mathinner{..} n] = I_1 I_2 \cdots I_\ell$ (the concatenation of $I_1, I_2, \ldots, I_\ell$ is equal to $[1 \mathinner{..} n]$),
(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) $O(n^3 u)$ time.
(C) $O(n^2 u)$ time.
(D) $O(n^3 k)$ time.
(E) $O(n^2 k)$ time.

---

21. (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.

---

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

---

23. (1 point) All problems in NP are solvable in exponential time. This statement is

(A) False.
(B) True.

24. (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) $O(n^2 + m^2)$ time.
(B) $O(2^n - 2^m)$ time.
(C) This is Satisfiability and it can not be solved in polynomial time unless $P = NP$.
(D) $O(n \log n + m)$ time.
(E) $O(n + m)$ time.

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

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

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

28. (3 points) Consider a CNF formula F with m clauses and n variables. The problem of computing an assignment that satisfies as many clauses as possible, is

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

(B) Can be solved in linear time.

(C) NP-Hard.

(D) None of the other answers are correct.

---

29. (3 points) Consider the problem of checking if a graph has a Hamiltonian path in it. This problem can be solved in

(A) None of the other answers are correct.

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

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

(D) Polynomial time.

---

30. (3 points) For the language $L = \{a^n b^n c^n \mid n \geq 0\}$, we have

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

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

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

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

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

---

31. (2 points) Consider a Turing machine (i.e., program) M that accepts an input $w \in \Sigma^*$ if and only if there is a DFA that accepts w. Then the language of $L(M)$ is

(A) context-free.

(B) finite.

(C) not well defined.

(D) Σ^*.

(E) undecidable.

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

(A) This would imply that $P = NP$.
(B) This would imply that $P \neq NP$.
(C) Impossible since $P \neq NP$.
(D) One can in polynomial time output if G has a an vertex cover of size $\leq k$ or not.
(E) At least two of the other answers are correct.

---

33. (3 points) Let G be a directed graph with weights (which can be either positive or negative). The graph G has n vertices and m edges. Computing the *longest* **simple** path between two vertices in G can be done in:

(A) None of the above.
(B) This can be solved in $O(nm)$ time using Bellman-Ford.
(C) This is NP-Hard.
(D) This is not defined if there are negative cycles in the graph. As such, it can not be computed.
(E) This can be solved in $O(n \log n + m)$ time using Dijkstra.

---

34. (2 points) Consider a DFA M with m states defined over $\{0, 1\}^*$. There is an equivalent regular expression r (i.e., $L(r) = L(N)$), such that (tighter is better):

(A) r is of length at most $f(m)$, where f is some function that is not specified in the other answers.
(B) r is of length at most $O(m)$.
(C) r is of length at most $O(m \log m)$.
(D) none of other answers.

---

35. (3 points) If a problem is NP-Hard, then it can also be undecidable. This statement is

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

36. (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) Undecidable.

(B) Decidable.

(C) Regular.

(D) Context-free.

(E) Finite.