Algorithms – II

Fourth Short Test

16-Nov-2020

---

1. Let A be a ρ-approximation algorithm for a maximization problem Q. Let OPT be the optimal solution for an instance I of Q, and SUBOPT the solution produced by A on the same input instance. Which of the following statements is necessarily *false*?

(A) SUBOPT = OPT
(B) SUBOPT = $\rho\times$ OPT
$\boxed{\textbf{(C)}}$ SUBOPT $< \rho\times$ OPT
(D) SUBOPT $> \rho\times$ OPT

2. We run the matching-based 2-approximation algorithm for computing a minimum vertex cover of the complete graph K_n on $n \geqslant 100$ vertices. Which of the following statements is necessarily true?

(A) The algorithm produces a vertex cover of size twice the optimal.
(B) The algorithm produces an optimal vertex cover.
(C) The algorithm produces a non-optimal vertex cover.
$\boxed{\textbf{(D)}}$ The algorithm produces a vertex cover with at most one more vertex than in an optimal cover.

3. Let TSP denote the general traveling salesperson problem for n cities (where we have any positive real-valued weight function on the edge set). Let τ denote the ratio of the largest edge weight w_{max} to the smallest edge weight w_{min}. Which of the following statements is true about TSP?

$\boxed{\textbf{(A)}}$ TSP can have a τ-approximation algorithm.
(B) TSP can have a w_{min}-approximation algorithm.
(C) TSP can have a w_{max}-approximation algorithm.
(D) TSP can have a $(\log n)$-approximation algorithm.

4. Let A be a fully polynomial-time approximation scheme for solving an optimization problem. With the notations n and ε having the usual meanings, which of the following can be a running time for A?

(A) $O(e^{\frac{1}{\varepsilon}\log_2 n})$ $\boxed{\textbf{(B)}}$ $O(e^{\log_2 \frac{n}{\varepsilon}})$ **(C)** $O(n^{\log_2 \frac{1}{\varepsilon}})$ **(D)** $O((\log_2 n)^{\frac{1}{\varepsilon}})$

5. There are n persons applying for n jobs in a company. The salary for Person i for Job j is s_{ij}. The company wants to appoint the persons to the jobs such that each person gets exactly one job, and the total salary to be paid is as small as possible. We want to formulate this problem as a 0,1-valued linear-programming problem. To that effect, we introduce the variables x_{ij} with the implication that $x_{ij} = \begin{cases} 1 & \text{if Person } i \text{ gets Job } j, \\ 0 & \text{otherwise.} \end{cases}$ Which of the following correctly describes the LP formulation for the given problem?

(A) Minimize $\sum\limits_{i,j} s_{ij}x_{ij}$ subject to $\sum\limits_{i,j} x_{ij} = n$.
(B) Minimize $\sum\limits_{i,j} s_{ij}x_{ij}$ subject to $\sum\limits_{i} x_{ij} = 1$ for all j.
(C) Minimize $\sum\limits_{i,j} s_{ij}x_{ij}$ subject to $\sum\limits_{j} x_{ij} = 1$ for all i.
$\boxed{\textbf{(D)}}$ Minimize $\sum\limits_{i,j} s_{ij}x_{ij}$ subject to $\sum\limits_{i} x_{ij} = 1$ for all j, and $\sum\limits_{j} x_{ij} = 1$ for all i.

6. Consider a Monte Carlo algorithm for primality testing based only on Fermat witness of compositeness, with the error probability bounded tightly by x. What is the maximum possible value of x?

Solution 1 (Think of Carmichael numbers)

7. Consider Karger's min-cut algorithm being run on a graph G with 22 nodes and 31 edges that has a single minimum cut. Which of the following is the tightest lower bound on the probability that the minimum cut is still present after 5 steps?

(A) 0.66 $\boxed{\textbf{(B)}}$ 0.58 **(C)** 0.53 **(D)** 0.49

8. Consider running the Karger–Stein algorithm on a graph with 24 nodes and 77 edges. Which of the following is the closest to the number of times the algorithm will be called (the initial call + the recursive calls)? Use the algorithm given in class. In your computations, whenever you have a non-integer and you have to make it an integer, choose ceiling. Assume that the minimum cut is directly computed when 6 or less nodes are left.

(A) 16 **(B)** 25 $\boxed{\textbf{(C)}}$ 31 **(D)** 63

9. Which of the following statements is true?

(A) Any Monte Carlo algorithm for a decision problem with one-sided error probability p and time complexity T can be converted into a Las Vegas algorithm with expected time complexity $O(n^k T)$, where n is the input size of the problem, and k is some constant $\geqslant 0$.

(B) Any Las Vegas algorithm with expected running time T can be converted into a Monte Carlo algorithm with running time pT and error probability $\leqslant p$ for some $0 < p \leqslant 1$.

$\boxed{\textbf{(C)}}$ There exists a Monte Carlo algorithm for some optimization problem with error probability p and time complexity T, that can be converted into a Las Vegas algorithm with expected time complexity $O(n^k T)$, where n is the input size of the problem, and k is some constant $\geqslant 0$.

(D) Any Monte Carlo algorithm for a decision problem with two-sided error can always be converted into a Monte Carlo algorithm for the same problem with one-sided error with a higher error probability.

10. Consider a Bloom Filter with size 6×10^6, and we insert 10^6 entries in it first. The filter uses 4 hash functions. What is the probability of a false positive if an element is searched for now?

Solution 0.0561 (numerical answer, set correct-answer range to $[0.055, 0.057]$)