CECS 528, Learning Outcome Assessment 11, 4/30/2025, Dr. Ebert

Problem

LO7. An instance of the `Load Balancing` decision problem is an array a of size n, a positive integer $p \geq 2$, as well as a positive integer k. The problem is to decide if the members of a can be partitioned into p different subarrays $b_1, \ldots, b_p$ so that the sum of the members in any one of the arrays does not exceed k. Our goal is to prove that `Load Balancing` is an `NP` problem.

(a) For a given instance (a, p, k) of `Load Balancing`, describe a certificate in relation to (a, p, k). Hint: a single array will suffice.

(b) Provide a semi-formal verifier algorithm that takes as input i) an instance (a, p, k) of `Load Balancing`, ii) a certificate for (a, p, k) as defined in part a, and decides if the certificate is valid for (a, p, k).

(c) Clearly define size parameters for the `Load Balancing` decision problem.

(d) Use the size parameters from part c to describe the running time of your verifier. Defend your answer.

LO8. Answer the following.

(a) Consider the mapping reduction $f : \texttt{3SAT} \rightarrow \texttt{Clique}$ from `3SAT` to `Clique` that was presented in lecture and the `3SAT` instance

$$\mathcal{C} = \{c_1 = (x_1, x_2, x_2), c_2 = (\overline{x}_1, \overline{x}_2, \overline{x}_2), c_3 = (\overline{x}_1, x_2, x_2), c_4 = (x_1, \overline{x}_2, \overline{x}_2)\}.$$

i. Compute $f(\mathcal{C}) = (G, k)$. Make sure to draw the entire graph.

ii. Verify that f is valid for input $\mathcal{C}$ from part i) in the sense that both $\mathcal{C}$ and $f(\mathcal{C})$ are either both positive instances or both negative instances. Make sure your answer is specific to the particular instances from part i).

(b) Consider the mapping reduction $f : \texttt{3SAT} \rightarrow \texttt{Subset Sum}$ from `3SAT` to `Subset Sum` that was presented in lecture. Draw the complete table associated with $f(\mathcal{C}) = (S, t)$ and either provide a subset $A \subseteq S$ that sums to t or explain why one does not exist.

LO9. Do the following.

(a) Demonstrate the `Vertex Cover` approximation algorithm described in lecture when applied to the graph $G = (V, E)$ whose edges are

$$E = \{(0,5), (0,9)(1,6), (2,5), (2,7), (2,3), (3,4), (3,6), (3,9), (1,4), (1,5),$$

$$(5,8), (3,7), (4,8), (8,9), (0,10), (7,10)\}$$

Clearly indicate how each vertex was selected for the cover.

(b) Answer the following regarding the analysis of the `k-Clustering` approximation algorithm.

i. In the analysis of the k-`Clustering` approximation algorithm recall that point x was introduced as what would have been the next center (assuming $k+1$ or more desired clusters). How is r defined in relation to x and how do we know that, for any two centers c_1 and c_2, $d(c_1, c_2) \geq r$?

ii. For any non-center y that is in some cluster with center c, how do we know that $d(y, c) \leq r$?

iii. How do we know that the optimal solution must have at least one cluster with a diameter of at least r?

LO10. Solve the following and show all work.

(a) A bag consists of four coins. Two of the coins are two-headed, one of the coins is one-headed, and the fourth coin is zero-headed. If two of the coins are randomly selected from the bag and are randomly tossed on to a table, determine the probability that both will show a head. Show work.

(b) For the experiment described in part a) determine the domain and probability distribution for the random variable $E[H|S]$, where H is the number of heads that appear on the table, and S is the number of two-headed coins that were selected.

(c) For the experiment described in part a), how many times should we expect to have to repeat the experiment before we witness no heads appearing? Show work and explain.

LO11. Answer the following questions about Karger's algorithm.

(a) A graph G has edges

$$\{e_1 = (1,4), e_2 = (4,6), e_3 = (1,2), e_4 = (2,5), e_5 = (2,3),$$

$$e_6 = (3,5), e_7 = (3,6), e_8 = (5,6), e_9 = (1,3)\}.$$

Draw the graph that remains after edges e_6, e_9, and e_4 have been selected in rounds 1, 2, and 3 of Karger's algorithm, respectively.

(b) Regarding part a), provide the probability of not selecting a min-cut edge in Round 1. Do the same for Round 2, conditioned on e_6 being selected in Round 1. Do the same for Round 3, conditioned on e_6 and e_9 being selected in Rounds 1 and 2, respectively, and finally for Round 4 assuming that a non-min-cut edge is selected in this final round. Compare the product of these four probabilities with the lower-bound probability p (as stated in Karger's Theorem) that a single execution of Karger's algorithm will return G's min cut.

(c) When proving the lower-bound probability that Karger's algorithm finds a minimum cut, the proof heavily relies on a property that is posssessed by every vertex of the input graph during each round of the algorithm, assuming the algorithm still has a chance of returning the minimum cut. State this property.