CECS 419-519, Writing Assignment 6, Due 8:00 am, March 22nd, 2024, Dr. Ebert

Directions

Make sure name is on all pages. Order pages (front and back) so that solutions are presented in their original numerical order. **Please no staples or folding of corners (your papers won't get lost). A paper clip is OK** Show all necessary work and substantiate all claims. Avoid plagiarism.

Problems

1. Solve the following.

   (a) Given a probability distribution with n probabilities, Huffman's algorithm may be applied to obtain an optimal average-length ternary code over the alphabet $\{0, 1, 2\}$, quaternary code over $\{0, 1, 2, 3\}$, and, in general, an optimal b-ary code over the alphabet

   $$\{0, 1, \ldots, b - 1\}.$$

   When going from base 2 to base b, the algorithm undergoes two changes: i) in Steps 2 and beyond, the b least probabilities are merged to form a new probability that is equal to their sum, and ii) in Step 1, the k least probabilities are merged, $2 \leq k \leq b$, where k is chosen so that, in all subsequent steps, exactly b least probabilities are merged until there is only a single probability (namely 1.0) remaining. Provide a formula for computing k. Hint: express your answer as a congruence between two quantities modulo some third quantity, i.e. of the form $X \equiv Y \bmod Z$, for some values of X, Y, and Z that depend on n, b, and k. (10 pts)

   Solution. If we apply Huffman's algorithm to n probabilities to create an optimal b-ary code, then the first merge will reduce the number of probabilities by $k - 1$. Then for each subsequent merge the number of probabilities gets reduced by $b - 1$. Letting s be the number of subsequent steps, we thus have

   $$k - 1 + (b - 1)s = n - 1,$$

   which is equivalent to

   $$k \equiv n \bmod (b - 1). \quad \square$$

(b) Apply your formula from part a to obtain an optimal average-length ternary code for the probability distribution $P = \{0.3, 0.3, 0.15, 0.1, 0.05, 0.05, 0.03, 0.02\}$. Verify that your initial reduction size k satisfies the equation from part a. (10 pts)

Solution. Since $b - 1 = 2$ and $n = 8$, we must have $k = 2$ and so we initially merge the two probabilities 0.02 and 0.03, and merge the three least probabilities in steps 2 through 4. The following is an optimal code:

$$0.3 : 0,\ 0.3 : 1,\ 0.15 : 20,\ 0.1 : 21,\ 0.05 : 220,\ 0.05 : 221,\ 0.03 : 2220,\ 0.02 : 2221.$$

(c) Compute the average codeword length for the code you obtained in part b and show that it is within 1 of the $H_3(P)$, the base-3 entropy of P. (5 pts)

Solution. The average length of the code from part b is 1.6, compared to $H_3(P) = 1.52$, and so the two values are within 0.8 which is consistent with Shannon's Noiseless Coding theorem.

2. An instance of the `Max Cut` decision problem is a simple graph $G = (V, E)$ and an integer $k \geq 0$. The problem is to decide if there is a way to color the vertices of G using the colors red and blue and results in there being at least k edges $e = (u, v)$ for which u and v are assigned different colors.

   (a) For a given instance (G, k) of `Max Cut`, describe a certificate in relation to (G, k). (5 pts)

   (b) Provide a semi-formal verifier algorithm that takes as input i) an instance (G, k) of `Max Cut`, ii) a certificate for (G, k) as defined in part a, and decides if the certificate is valid for (G, k). (10 pts)

   (c) Use the size parameters $m = |E|$ and $n = |V|$ to describe the running time of your verifier from part b. Defend your answer in relation to the algorithm you provided for the verifier. (5 pts)

   Solution. See problem LO2 from the CECS 329 March 14th, 2024 Learning Outcome Assessment.

   `https://home.csulb.edu/~tebert/teaching/spring24/329/assess/`
   `LO6-03-14-2024/LO6-03-14-2024-Solutions.pdf`

3. Answer the following.

   (a) For the mapping reduction f : `Subset Sum` $\rightarrow$ `Set Partition` defined in lecture, determine $f(S, t)$ for `Subset Sum` instance $(S = \{2, 6, 8, 9, 12, 14, 18, 26, 30, 33\}, t = 64)$. Show work. (5 pts)

   Solution. We have

$$M = \sum_{s \in S} s = 158$$

   with $t = 64 < M/2 = 79$. Thus, $f(S, t) = S + \{J\}$, where

$$J = 158 - 2t = 158 - 128 = 30.$$

(b) Verify that both (S, t) and $f(S, t)$ are both positive instances of their respective decision problems. Provide solutions for both instances. (10 pts)

Solution. (S, t) is a positive instance of SS since $A = \{2, 14, 18, 30\}$ sums to $t = 64$. Moreover, $f(S, t)$ is a positive instance of SP since the members of $A \cup \{J\}$ sum to 94, as do the members of

$$B = \overline{A \cup \{J\}} = \{6, 8, 9, 12, 26, 33\}. \quad \square$$

4. Given 3SAT instance

$$\mathcal{C} = \{(x_1, \overline{x_3}, \overline{x_5}), (x_1, x_3, \overline{x_4}), (x_1, x_2, \overline{x_3}), (x_1, x_3, x_5), (\overline{x_1}, x_2, x_3), (\overline{x_1}, x_2, \overline{x_4}),$$

$$(\overline{x_1}, \overline{x_2}, x_4), (\overline{x_1}, \overline{x_2}, \overline{x_5}), (\overline{x_1}, x_3, x_5), (\overline{x_1}, \overline{x_3}, x_4), (x_2, x_4, \overline{x_5}),$$

$$(\overline{x_2}, \overline{x_3}, x_5), (\overline{x_2}, x_3, \overline{x_4})(x_2, x_3, \overline{x}_4), (x_1, x_2, x_4), (\overline{x}_1, \overline{x}_3, \overline{x}_4)\},$$

answer the following questions.

(a) For the mapping reduction f from 3SAT to `Clique` presented in lecture, if $f(\mathcal{C}) = (G, k)$, then how many vertices does G have? How many edges does it have? Explain and show work. Hint: there is a clever way for calculating the number of edges without having to enumerate them. What is the value of k? (10 pts)

Solution. Since $m = |\mathcal{C}| = 16$, G has $16 \times 3 = 48$ vertices. To calculate its number of of edges, note that there are at most $9 \binom{16}{2} = 1080$ possible edges for a reduction involving 16 clauses. However, we must subtract from this upper bound any "forbidden edge", i.e. one that would connect a literal with its negation. Moreover, for a given variable x, the number of such edges equals the product $p \times n$ where p (respectively, n) is the number of positive (respectively, negative) occurrences of the variable in any of the clauses. Thus, the number of forbidden edges induced by each variable is

$$x_1 : 5 \times 7 = 35,\ x_2 : 6 \times 4 = 24,\ x_3 : 6 \times 5 = 30,\ x_4 : 4 \times 5 = 20,\ x_5 : 3 \times 3 = 9$$

for a total of 118 forbidden edges. Therefore, G has $1080 - 118 = 962$ edges.

(b) Does G have a k-clique? If yes, provide the vertices of the clique (for clarity make sure to indicate the vertex group for each vertex). In any case, defend your answer. (10 pts)

Solution. Since $\alpha = (0, 1, 0, 0, 1)$ satisfies $\mathcal{C}$, it follows that G has a clique size equal to $k = m = 16$, where the vertex used in vertex group (clause) i is the first vertex labeled with a literal that is set to 1 by α (such a set of literals must be connsistent, since an assignment cannot set both a variable and its negation to 1). Thus, in the order of vertex groups $c_1, \ldots, c_{16}$ the clique is

$$C = \{\overline{x_3}, \overline{x_4}, x_2, x_5, \overline{x_1}, \overline{x_1}, \overline{x_1}, \overline{x_1}, \overline{x_1}, \overline{x_1}, x_2, \overline{x_3}, \overline{x_4}, x_2, x_2, \overline{x_1}\}.$$

5. Suppose you have acccess to a polynomial-time algorithm $\mathcal{A}$ that decides the `Set Partition` problem. Use $\mathcal{A}$ to devise a polynomial-time algorithm that computes the partitioning set $A \subset S$ for any positive instance S of SP. Please do your best to carefully explain the key idea(s) behind your algorithm before reverting to (sigh!) pseudocode. Every problem has at least one "key idea" in its solution. Start with the key idea and *not* the boring details. (20 pts)

Solution. Let $S = \{s_1, s_2, \ldots, s_n\}$ be a positive intance of SP with $n \geq 2$. If $n = 2$ then $A = \{s_1\}$ is the partitioning set. So assume $n \geq 3$, and, for all $2 \leq k < n$ we have a method for computing a partitioning set for a positive instance of SS that has cardinality k. Consider $a_1 \in S$. It either belongs in a partition set A for S with at least one other member, or it partitions S alone in a singleton set. The latter case can be easily checked. In the former case, for each $i \in \{2, \ldots, n\}$, we may temporarily remove s_1 and s_i from S and replace them with $s_1 + s_i$. Then letting

$$S' = S - \{s_1, s_i\} + \{s_1 + s_i\},$$

if $\mathcal{A}(S') = 1$, then, since $|S'| = n - 1 < n$, we may compute a partition set A' for S' and, if $s_1 + s_i \notin A'$, then $A = A'$ is a partition set for S. Otherwise,

$$A = A' - \{s_1 + s_i\} + \{s_1, s_i\}$$

is the desired partition set. On the other hand, if $\mathcal{A}(S') = 0$, then we increment i and repeat the process until either a partition set has been identified, or we conclude that $\{a_1\}$ is itself a partitioning set.

Finally, letting $T(n)$ denote the running time for a positive instance of size n, we see that $T(n)$ satisfies the recurrence

$$T(n) = T(n-1) + \mathrm{O}(n \cdot q(n)),$$

where $q(n)$ is a polynomial that represents the running time of $\mathcal{A}$. Notice that we must have $\mathrm{O}(n \cdot q(n))$, since at each level of recursion, we must call $\mathcal{A}$ at most n times. Finally, since there are at most n levels of the recursion and the most time-intensive level is level-0, we see that the total running time will not exceed $\mathrm{O}(n^2 \cdot q(n))$. □