Lab10-Approximation & Randomized Algorithm

CS214-Algorithm and Complexity, Xiaofeng Gao, Spring 2019.

∗ If there is any problem, please contact TA Mingran Peng.
∗ Name: KylinChen Student ID: 517030910155 Email: k1017856853@icloud.com

1. Given a CNF Φ with n boolean variables $\{x_i\}_{i=1}^n$ and m clauses, with each clause consisting of 3 boolean variables. For example $\Phi = C_1 \wedge C_2 = (x_1 \vee \overline{x_2} \vee \overline{x_4}) \wedge (\overline{x_1} \vee \overline{x_2} \vee \overline{x_3})$. Assume that Φ is satisfiable, the goal is to find the feasible assignment of $\{x_i\}_{i=1}^n$ with **fewest true boolean variables**.

   (a) Please formulate it into integer programming.

   (b) Design an approximation algorithm based on deterministing rounding. Choose its approximation ratio and explain. Pseudo code is needed.

2. (Bonus)Suppose there is a sequence of pearls of different color. Color is denoted as $1 - m$ and the total number of pearls is n. After you read these information and conduct some pre-processing, you need to face lots of of queries.

   A query gives two positions $1 \leq l \leq r \leq n$, and ask whether there exists a color, that at least half of pearls in $[l, r]$ is such color.

   (a) Design a random algorithm to solve this problem. Space complexity of your algorithm should be strictly better than $O(mn)$. Explain your idea briefly, give time complexity for pre-processing and per query, and give space complexity. Your accuray should be better than 99.9%.

   For example, a naive algorithm just read in all pearls as pre-processing. And naively iterate every color and every postion for query. This case, the pre-processing complexity is $O(n)$. For query, it will execute $(r-l) * m$ times, since $r-l$ can achieve $n-1$, so time complexity per query is $O(mn)$. No extra space needed.

   (Hint: Random choose some color and examine.)

   (b) **Remark:** This question involves a little bit knowledge about online algorithm. The ddl for this lab is 5/27/2019.

   Now there are extra operation besides query.

   Append(c): Put a peral with color c at the end of sequence.

   Erase: Take out the last pearl.

   Colouration(p,c): Choose pearl of postion p and change its color to c.

   Assume that no operation will involve a new color. You may modify your algorithm and show time complexity for each type of operation(include query).

   (Hint: Consider Balanced Binary Tree. Given an element e, they can find whether e exists in tree, and how many elements in tree are smaller than e, in $O(logn)$ time.)

Remark: You need to include your .pdf and .tex files in your uploaded .rar or .zip file.