ShanghaiTech University

CS240 Algorithm Design and Analysis

Spring 2021

Problem Set 5

Due: 23:59, May 25, 2021

1. Submit your solutions to Gradescope (www.gradescope.com).
2. In "Account Settings" of Gradescope, set your FULL NAME to your Chinese name and enter your STUDENT ID correctly.
3. If you want to submit a handwritten version, scan it clearly. Camscanner is recommended.
4. When submitting your homework, match each of your solution to the corresponding problem number.

Problem 1:

As praise for passing his exam, Bobby received from his parents n coins, where the i'th coin has a *size* $c_i > 0$. Bobby wants to store these coins in a bunch of piggy banks. All the piggy banks have size V, where $V \geq \max_i c_i$. To minimize the number of piggy banks he uses, Bobby plans to do the following: He starts with one *active* piggy bank. Then Bobby goes through the coins one by one, and tries to put each coin in any active piggy bank in which it fits. If the coin doesn't fit in any active piggy bank, Bobby opens a new active piggy bank. The algorithm is shown below. Prove that this algorithm is a 2-approximation, *i.e.* that Bobby will use at most twice the minimum possible number of piggy banks.

Input: Size of coins $c_1, c_2, \cdots, c_n$; size of piggy bank V

$b \leftarrow 1$ ▷ current number of active piggy banks

$P_1, P_2, \ldots \leftarrow 0$ ▷ P_i = amount of space used in i'th piggy bank; initially all piggy banks are empty

for $i = 1$ to n **do**

    if $\exists j \leq b$: $P_j + c_i \leq V$ **then**

        $j \leftarrow$ any j with $P_j + c_i \leq V$

        $P_j \leftarrow P_j + c_i$ ▷ put c_i in j'th active piggy bank

    else

        $b \leftarrow b + 1$ ▷ open a new piggy bank and put c_i in it

        $P_b \leftarrow c_i$

Problem 2:

Recall that the maximum independent set problem for a graph $G = (\mathcal{V}, \mathcal{E})$ is to find a set $\mathcal{T} \subseteq \mathcal{V}$ such that for all $u, v \in \mathcal{T}$, both $(u, v), (v, u) \notin \mathcal{E}$. The goal is to maximize $|\mathcal{T}|$, i.e. the number of vertices in $\mathcal{T}$.

Here is a possible algorithm for this problem:

1. Set $\mathcal{T} = \emptyset$.
2. Iterate through the vertices in $\mathcal{V}$ in a random order.
3. When processing a vertex v (*i.e.* we have already processed all vertices earlier than v in the random ordering), if none of v's neighbors have been added to $\mathcal{T}$ yet, add v to $\mathcal{T}$

4. return $\mathcal{T}$.

Show that this algorithm produces a $\frac{1}{d+1}$-approximation, where d is the maximum degree of any vertex. Also, describe how to implement the algorithm efficiently.

Problem 3:

A group of n children are arranged in a circle. Some of the children have secrets they want to share with others. Denote a secret by (i, j), indicating child i wants to share a secret with child j. Secret (i, j) is written on a note and is passed around the circle from child to child, starting from i and ending at j. The note can be passed either in the clockwise or counterclockwise direction. Let the number of notes passed between children i and $i+1 \pmod{n}$ (in either direction) be T_i, and let $T = \max_i T_i$ be the maximum number of notes passed between any two neighboring children. Give a 2-approximation algorithm to minimize T and prove your algorithm is correct.