ShanghaiTech University

# CS240 Algorithm Design and Analysis

## Fall 2023

## Problem Set 4

Due: 23:59, Jan. 19, 2024

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:

If the set of stack operations included a MULTIPUSH operation, which pushes $k$ items onto the stack. Analyze the amortized cost of stack operations (including PUSH, POP, MULTIPOP and MULTIPUSH).

```
MULTIPUSH(S, a, k)
    While k > 0
        PUSH(S, a[k])
        k = k - 1
```

## Problem 2:

Suppose we perform a sequence of $n$ operations on a data structure in which the $ith$ operation costs $i$ if $i$ is an exact power of 3, and 1 otherwise. Use aggregate analysis to determine the amortized cost per operation.

## Problem 3:

Given a set of positive integers, $A = a_1, a_2, ..., a_n$. And a positive integer B. A subset $S \in A$ is GOOD if

$$\sum_{a_i \in S} a_i \leq B$$

Given an approximation algorithm that it returns a GOOD subset whose total sum is at least half as large as the maximum total sum of any GOOD subset, with the running time at most $O(nlogn)$

## Problem 4:

An undirected graph G = (V, E) with node set V and edge set E is given. The goal is to color the edges of G using as few colors as possible such that no two edges of the same color are incident to a common node. Let OPT(G) denote the minimum number of different colors needed for coloring the edges of G.

Show that there exists a Greedy algorithm that needs at most $2 \cdot \text{OPT(G)}$-1 different colors for any graph G. Prove that your algorithm always obtains a valid solution, i.e., no two edges of the same color are incident to a common node

## Problem 5:

Given a function rand2() that returns 0 or 1 with equal probability, implement rand3() using rand2() that returns 0, 1 or 2 with equal probability. Minimize the number of calls to rand2() method. Prove the correctness.

## Problem 6:

Assume that you have a function `randM()` which returns an integer between 0 and $M-1$ (inclusive) with equal probability. Write an algorithm using the `randM()` function to implement a `randN()` function, where $N$ is not necessarily a multiple of $M$, but `randN()` needs to return an integer between 0 and $N-1$ with equal probability.