ShanghaiTech University

# CS240 Algorithm Design and Analysis

Spring 2023

Problem Set 1

Due: 23:59, Mar. 6, 2023

1. Submit your solutions to the course Blackboard.
2. If you want to submit a handwritten version, scan it clearly.
3. Late homeworks submitted within 24 hours of the due date will be marked down 25%. Homeworks submitted more than 24 hours after the due date will not be accepted unless there is a valid reason, such as a medical or family emergency.
4. You are required to follow ShanghaiTech's academic honesty policies. You are allowed to discuss problems with other students, but you must write up your solutions by yourselves. You are not allowed to copy materials from other students or from online or published resources. Violating academic honesty can result in serious penalties.

## Problem 1:

Assume you have functions $f$ and $g$ such that $f(n) = O(g(n))$. For each of the following statements, decide whether you think it is true or false and give a proof or counterexample.

a) $\log_2 f(n) = O(\log_2 g(n))$

b) $2^{f(n)} = O(2^{g(n)})$

c) $f(n)^2 = O(g(n)^2)$

## Problem 2:

Sort the following functions in ascending order of growth.

$$f_1(n) = (\log n)^{\log n} \tag{1}$$
$$f_2(n) = n^2 \log n \tag{2}$$
$$f_3(n) = 2023^{\sqrt{n}} \tag{3}$$
$$f_4(n) = n^{987} \tag{4}$$
$$f_5(n) = 2^n \tag{5}$$
$$f_6(n) = (\log n)^n \tag{6}$$
$$f_7(n) = 16^{42^{223}} \tag{7}$$
$$f_8(n) = n \tag{8}$$
$$f_9(n) = n^{\sqrt{n}} \tag{9}$$

## Problem 3:

Consider the following method for sorting $N$ elements stored in an array $A$, called *selection sort*. First, find the smallest element in $A$ and swap it with element $A[1]$. Next, find the second smallest element in $A$ and swap it with the element $A[2]$. Continue this way for the first $N-1$ elements in $A$. Analyze its best and worst case time complexity using big-O notation.

## Problem 4:

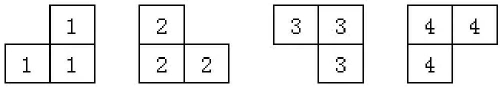
Suppose you have $n$ identical-looking cards. Each card has an ID, which you cannot see, and there may be multiple cards with the same ID. You are given a machine which can take two cards at a time and determine whether the cards have the same ID.

Give an algorithm which uses the machine $O(n \log n)$ times, and determines whether there exists an ID which occurs on more than $\lfloor n/2 \rfloor$ cards. For example, if the cards have IDs $\{1, 2, 2, 2\}$, then the algorithm should return yes, while for $\{1, 2, 3, 3\}$ it should return no. You can assume that $n$ is a power of 2.

## Problem 5:

Jack wants to use carpets to cover the floor of the Student Activity Center. However, he can only find small carpets with certain special shapes. He also needs to leave a given cell of the floor uncovered, which will be used for placing a bulletin board.

- The floor of the Student Activity Center is a square shape where each side has length $n$, giving a total area of $n^2$ units. You can assume $n = 2^k$ for some $k \geq 2$.
- There are four shapes of carpets, and each covers three unit cells of the floor. The shapes are shown below.
- The bulletin board will be put in one cell of the floor, and takes 1 unit of area.
- All other cells without the bulletin board need to be covered by exactly one carpet. In another word, two carpets cannot overlap.



As simple example, suppose the floor is a $2 \times 2$ square, and the bulletin board is placed in cell $(0, 0)$. Then we can use one carpet with shape 1 to cover the floor.

Given an algorithm for Jack to solve the problem with an $n \times n$ floor, where the bulletin board is placed in cell $(x, y)$, for $0 \leq x, y \leq n - 1$. Clearly describe your algorithm and why it is correct, and use pseudocode if necessary.

*Hint*: Using divide-and-conquer, and consider the problem for a $4 \times 4$ floor first.

## Problem 6:

Suppose you have $n$ identical-looking integrated-circuit chips, each of which may be good or bad (i.e. work correctly or not). To test the chips, you have a machine where you can place two chips at a time, and then each chip reports whether it thinks the other chip is good or bad. A good chip always accurately reports whether the other chip is good or bad, but a bad chip can give any answer. Thus, the four possible outcomes each time you test a pair of chips are as follows:

| Chip A says | Chip B says | Conclusion |
|---|---|---|
| B is good | A is good | both are good, or both are bad |
| B is good | A is bad | at least one is bad |
| B is bad | A is good | at least one is bad |
| B is bad | A is bad | at least one is bad |

Give an algorithm to find a single good chip from among the $n$ chips, assuming that more than $\lfloor n/2 \rfloor$ of the chips are good. Use as few tests as you can.

*Hint*: Show that $\lceil n/2 \rceil$ tests are sufficient to reduce the problem to one of at most half the size.