National University of Singapore

School of Computing

IT5003 - Data Structures and Algorithms

Midterm Test

(AY2025/26)

Time Allowed: 70 minutes

INSTRUCTIONS TO CANDIDATES:

1. Do **NOT** open this assessment paper until you are told to do so.
2. This assessment paper contains just ONE section.
   It comprises SEVEN (7) printed pages, including this page.
3. This is an **Open Book Assessment**.
   You cannot use any electronic device except one non-programmable calculator.
4. You can use either pen or pencil. Just make sure that you write **legibly**!
5. Important tips: Pace yourself! Do **not** spend too much time on one (hard) question.
   Read all the questions first! Some (sub-)questions might be easier than they appear.
6. You can use **pseudo-code** in your answer but beware of penalty marks for **ambiguous answer**.
   You can use **standard, non-modified** classic algorithm in your answer by just mentioning its name, e.g. run Merge Sort on list L, use Stack S, use Binary Min Heap $MinPQ$, etc.
7. The total marks is 50. All the best :)

A Application Questions ($3 \times 16 + 2 = 50$ marks)

A.1 Grid Problem (16 marks)

You are given a 2D Integer matrix *grid* of size $m \times n$ ($1 \leq n, m$), an integer array *limits* of length m ($0 \leq limits[i] \leq n$), and an integer k ($0 \leq k \leq min(n, sum(limits))$. Your task is to choose at most k elements from the matrix *grid*, given that the number of elements chosen from the i-th row of grid does not exceed $limits[i]$, so that you get the maximum possible sum. See Table 1 for examples.

No	Input matrix *grid*	*limits*	k	Output	Explanation
1	$[[7,7,7],[7,7,7]]$	$[1,3]$	2	14	Take any two 7s
2	$[[5,10,3,1]]$	$[3]$	2	15	Take $5 + 10 = 15$
3	$[[1,2,3],[4,5,6],[7,8,9]]$	$[3,2,1]$	2	15	Take $6 + 9 = 15$ from 2nd and 3rd rows
4	$[[22,1,33],[21,4,77]]$	$[2,3]$	3	132	Got it?

Table 1: A Few Examples for this Grid Problem

Design algorithm that solves this task and analyze its worst-case time complexities.
Your solution will be given marks according to Table 2 below.

Special Case	Comments	Marks
All integers in *grid* are equal	Like in Sample 1	7
$m = 1$	One row, like in Sample 2	8
General cases	But not the fastest known runtime	14
General cases	With the fastest known runtime	16
-	Penalty of wrong time complexity analysis	-2

Table 2: Scoring Scheme

A.2 A Linked List Task (16 marks)

You are given the `head` of a *Singly* Linked List (SLL). There are n ($1 \leq n$) (not necessarily distinct) integers in the SLL. Each integer $\in$ [1..99]. Visualize the SLL from the left (`head` on the leftmost) to the right (`tail` on the rightmost) with arrows going to the right, like in VisuAlgo. Your task is to delete every vertex that satisfy a certain property as shown in Figure 1 and 2. Note that you *cannot* create a new SLL, i.e., you need to delete what you need to delete directly in the given SLL. Return the `head` of the *updated* SLL.

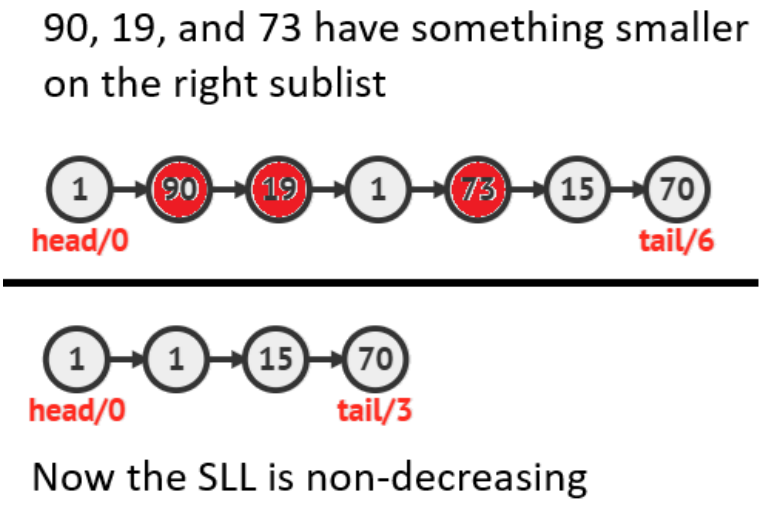


Figure 1: An example SLL and vertices to be deleted (top) and the updated SLL (bottom)

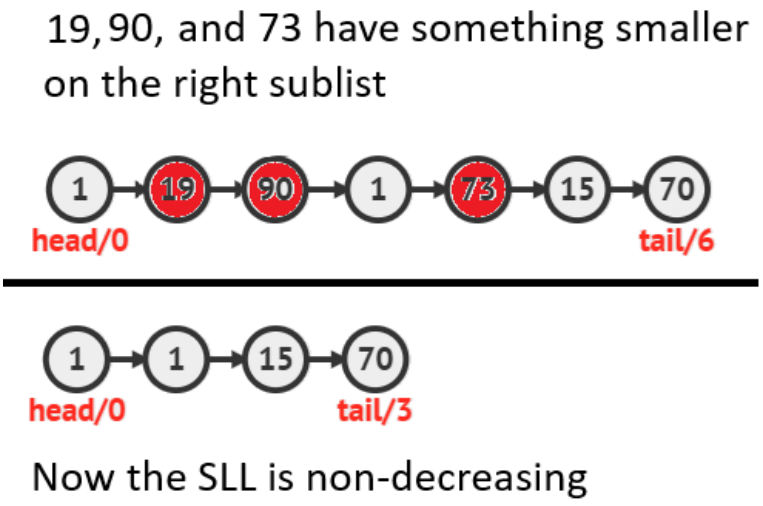


Figure 2: Another clearer example SLL and vertices to be deleted (top) and the updated SLL (bottom)

For full 16 marks, implement this SLL operations (use the given template at answer sheet).
Analyze the tightest worst-case time complexity of your solution, -5 marks if it is not optimal.
If your solution does not modify the SLL directly, you will be given -3 penalty.

A.3 Three Bags of Candies (16 marks)

Initially, you are given three bags of candies of *positive* sizes a, b, and c, respectively ($1 \leq a, b, c$). You repeatedly do this: Choose *any* two different *non-empty* bags, take one candy from each, and add 7 points to your score. You stop this process when there are fewer than two non-empty bags. Given three integers a, b, c, output the maximum possible score that you can get if you choose optimally.

No	Input a	b	c	Output	Explanation
1	1	3	4	28	Let's describe the number of candies left as tuple (a, b, c), $(1, 3, 4) \rightarrow (0, 3, 3) \rightarrow (0, 2, 2) \rightarrow (0, 1, 1) \rightarrow (0, 0, 0)$ There are other possible paths, but 28 is the maximum score
2	1	2	4	21	$(1, 2, 4) \rightarrow (0, 2, 3) \rightarrow (0, 1, 2) \rightarrow (0, 0, 1)$ We cannot use the last candy $c = 1$
3	7	7	1	49	Special Case 1
4	7	5	1	42	Special Case 2
5	2	3	5	35	General Case 1
6	2	2	2	21	General Case 2

Table 3: A Few Examples

Design an algorithm that solves this task.
and analyze its worst-case time complexities in terms of a, b, and/or c.
Your solution will be given marks according to Table 4 below.

Special Case	Comments	Marks
$a = b$ and $c = 1$	There is a very easy solution	4
$a \neq b$ and $c = 1$	An extension of the previous special case	8
General cases	But not the fastest known runtime	13
General cases	With the fastest known runtime	16
-	Penalty of wrong time complexity analysis	-2

Table 4: Scoring Scheme

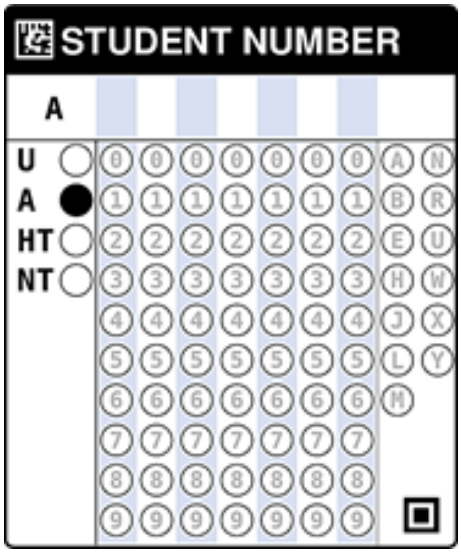
A.4 Midterm Feedback ($2 \times 1 = 2$ marks)

Prof Halim had run IT5003 for 10 iterations. For this S1 AY25/26, he is implementing these changes. So just focus on writing non-blank non-anonymous feedback on them:

- Does adding LeetCode – more concise problem description, closer to coding interview tasks, mostly for lecture demos (on top of Kattis – longer/more cryptic problem description from programming competition, mostly for programming assignments) helps you in understanding IT5003 contents so far? (PS: Compared to CS2040S that have additional Thursday morning sessions, IT5003 classes will only do TWO out of FIVE daily LeetCode weekly: Wednesday chosen LeetCode tasks and from Week 08 to 12, labs on Monday and Saturday (use the next Monday task) will discuss the chosen LeetCode tasks).
- What is your opinion about adding 10% midterm test (this paper) and reducing the weightage of machine-generated questions: VisuAlgo Online Quizzes (-10%) compared to the last 9 iterations of IT5003?

The Answer Sheet for Semester 1 AY2025/26

Write your Student Number in the box below using **(2B) pencil**. **Do NOT write your name**.



Section	Maximum Marks	Your Marks	Grading Remarks
Total	100		

Box A.1. Grid Problem

I claim that my solution runs in $O($_____$)$.

Box A.2. A Linked List Task

```
# Definition for singly-linked list.
# class ListNode:
#     def __init__(self, val=0, next=None):
#         self.val = val
#         self.next = next
class Solution:
    # you can create helper function(s)




    def removeNodes(self, head: Optional[ListNode]) -> Optional[ListNode]:
        # complete this




        # return the updated SLL at the end
```

I claim that my solution runs in $O($_____$)$.

Box A.3. Three Bags of Candies

I claim that my solution runs in $O($_____$)$.

Box A.4. Feedback about IT5003 S1 AY25/26 changes: Add LeetCode and Midterm

– END OF PAPER; All the Best –