National University of Singapore

School of Computing

IT5003 - Data Structures and Algorithms

Final Assessment

(Semester 1 AY2023/24)

Time Allowed: 2 hours

INSTRUCTIONS TO CANDIDATES:

1. Do **NOT** open this assessment paper until you are told to do so.
2. This assessment paper contains THREE (3) sections.
   It comprises SIXTEEN (16) printed pages, including this page.
3. This is an **Open Book Assessment**.
4. Answer **ALL** questions within the **boxed space** of the answer sheet (page 13-16).
   For Section A, shade the option in the answer sheet (use 2B pencil).
   There are a few starred (*) boxes: free 1 mark if left blank but 0 for wrong answer (no partial).
   The answer sheet is at page 13-16 but you will still need to hand over the entire paper.
   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 (subtask) 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 Dijkstra's on graph G, Kruskal's on graph G',
7. All the best :)

A MCQs ($20 \times 2 = 40$ marks)

Select the **best unique** answer for each question.
Each correct answer worth 2 marks.

Only Q1, 19, and 20 are archived. The rest are hidden to open up possibilities of reuse in the future.

1. Which of the following option is the correct and the fastest Python code to check if a given list `A` (not necessarily sorted) that has length n contains *two different* copies of integer 7. Assume that $2 \le n \le 100K$.

   a).
   ```
   contains_two_sevens = False
   for i in range(n):
       for j in range(n):
           if i == j:
               break
           if A[i] == 7 and A[j] == 7:
               contains_two_sevens = True
   print(contains_two_sevens)
   ```

   b).
   ```
   first_index = -1
   for i in range(n):
       if A[i] == 7:
           first_index = i
           break
   second_index = -1
   for j in range(n-1, 0, -1):
       if A[j] == 7:
           second_index = j
           break
   print(first_index != second_index)
   ```

   c).
   ```
   contains_two_sevens = False
   for i in range(n-1):
       if A[i] == A[i+1] == 7: # can Python do this?
           contains_two_sevens = True
   print(contains_two_sevens)
   ```

   d).
   ```
   first_index = A.index(7)
   # list.index(element, start, end)
   # if start is supplied, start searching from this index
   second_index = A.index(7, first_index+1)
   print(first_index != second_index)
   ```

   e). None of the above, each code has a subtle bug

[PAGE 3 IS NOT ARCHIVED]

[PAGE 4 IS NOT ARCHIVED]

[PAGE 5 IS NOT ARCHIVED]

[PAGE 6 IS NOT ARCHIVED]

[PAGE 7 IS NOT ARCHIVED]

19. For the Loophole in Subsection C.1.3, what data structure that we have learned in class that can correctly solve that problem in the fastest time complexity?

    a). Doubly Linked List (DLL)

    b). Python `deque`

    c). Python `heapq`

    d). Python `Counter`

    e). None of the above

20. For the Special Case in Subsection C.2.2, what is the best algorithm that we have learned in class that can correctly solve that problem in the fastest time complexity?

    a). DFS

    b). BFS

    c). Dijkstra's algorithm

    d). Topological Sort

    e). None of the above

B Simpler Questions (20 marks)

B.1 Another Nearly Sorted Variant (10 marks)

There are about 193 countries in the world who are the member states of the United Nations (this number can increase or even decrease in the future). We can sort these countries based on multiple criteria. For this question, let's deal with the following scenario:

We (treasurer) want to collect registration fees from $n = 87$ country delegations for a certain computing olympiad[1]. Most of these n delegation leaders are currently queueing in front of us in First-In First-Out (FIFO) order. The delegation leaders of these n countries can come in any of the $n!$ possible permutations (a lot). When a country leader at the head of the queue approaches us (treasurer), he/she pays the registration fee, mentions his/her country name, and expecting an official receipt with his/her country name and the treasurer's signature + stamp (for claim purposes later).

However, the list of receipts is currently sorted by the three-letter of these n country codes according to ISO 3166-1 alpha-3[2]. This seemingly random order of 9 specific country names: { 'Switzerland', 'Germany', 'Spain', 'United Kingdom', 'Croatia', 'Republic of Korea', 'Sri Lanka', 'El Salvador', 'South Africa'} is actually the sorted order of these 9 countries by their ISO 3166-1 alpha-3 codes: {'CHE', 'DEU', 'ESP', 'GBR', 'HRV', 'KOR', 'LKA', 'SLV', 'ZAF'}. $n = 87$ for this year. Fortunately, the other $87 - 9 = 78$ countries at least have their first character of ISO 3166-1 alpha-3 matching the *first* character of their country names, although we cannot assume the same for the second and/or the third characters, e.g., 'IDN' ('Indonesia') comes earlier than 'IND' ('India') although if we sort by full country names, 'India' should come before 'Indonesia'. Similar case with 'SGP' ('Singapore') versus 'SRB' ('Serbia').

B.1.1 Sort These Countries ($2 \times 2 = 4$ marks)

The following $n = 9$ countries are not currently sorted in any order: {'Indonesia', 'Sri Lanka', 'Slovenia' ('SVN'), 'Serbia', 'South Africa', 'India', 'Sweden' ('SWE'), 'Singapore', 'Slovakia' ('SVK')}. For 2 marks each, sort these n countries based on:

1. Their country names
2. Their ISO 3166-1 alpha-3 codes

B.1.2 What Are Your Advices? ($2 \times 3 = 6$ marks)

If you are the treasurer of that olympiad, what will you do to smoothen this process? Give two logical answers. Each answer will be graded with 0/1/2/3 marks for blank/wrong/illogical, has two or more issue(s), has an issue, or a very good advice, respectively.

B.2 Doubly Linked List with Duplicates (10 marks)

The current Doubly Linked List (DLL) visualization of VisuAlgo [3] does not allow duplicate yet. By next semester when this question is read as a past paper question, duplicates are probably allowed. For each of the 5 operation below, highlight what has to be modified, if any (2 marks each). See the Answer Sheet for the fill-in-the-blank questions.

[1] See `https://stats.ioinformatics.org/delegations/2023`
[2] See `https://en.wikipedia.org/wiki/ISO_3166-1_alpha-3`
[3] See `https://visualgo.net/en/list?mode=DLL` as of Thu, 7 Dec 2023.

C Applications (20+20 = 40 marks)

C.1 Marketing Promotion (20 marks)

There is a large supermarket that wants to run a specific marketing promotion (to increase sales):

- A customer who wants to participate in the promotion, writes his/her personal details on the receipt that he/she paid by him/herself, and throws it into a special ballot box.
- At the end of every day of the promotion, two receipts are taken out from the ballot box:

  1. First, the receipt amounting to the largest value is chosen (if ties, pick any (randomly)),
  2. Next, the receipt amounting to the smallest value is chosen (if ties, pick any (randomly));

  The customer who has paid the largest value gets a money prize equal to the difference between the value on his receipt and the value on the receipt amounting to the smallest value.

- To avoid multiple prizes for one purchase, both receipts selected according to the above rules are not returned to the ballot box, but all remaining receipts still participate in the promotion.

The turnover of the supermarket is very big, thus an assumption can be made, that at the end of every day, before taking out receipts amounting to the largest and the smallest values, there are at least 2 receipts in the ballot box.

You are given d ($1 \leq d \leq 7777$), the number of days of this promotion campaign and d more lines. Each of the next d lines describe one day. It starts with an integer k ($0 \leq k \leq 10^5$), the number of receipts generated that day, and then k positive integers describing the values on k receipts that day (all values are not more than 10^6 dollars). The total number of receipts (let's call this as variable n) throughout the entire d days will not exceed 10^6, i.e., $k_1 + k_2 + \ldots + k_d = n \leq 10^6$ and $d = O(n)$ for this problem.

Your task is to compute (based on the prices on receipts thrown into the ballot box on each day of promotion) what the total cost of prizes during the whole promotion will be.

For example: $d = 3$ days and these are the receipts thrown into the ballot box on those d days:

1. Day 1: $k_1 = 3$, receipts with values: $\{1, 2, 3\}$ are thrown into an initially empty ballot box.
   We have box = $\{1, 2, 3\}$ now.
   At the end of day 1, the one who paid 3 dollars get $3 - 1 = 2$ dollars prize.
   Then both 3 and 1 dollars receipts were removed, box = $\{2\}$ now.
2. Day 2: $k_2 = 2$, receipts with values $\{1, 1\}$ are thrown into a ballot box.
   As the box previously contains $\{2\}$, box = $\{1, 1, 2\}$ now.
   At the end of day 2, the one who pays 2 dollars get $2 - 1 = 1$ dollar prize.
   Then both 2 and 1 (either one) dollars receipts were removed, box = $\{1\}$ now.
3. Day 3: $k_3 = 4$, receipts with values $\{10, 5, 5, 1\}$ receipts are thrown into a ballot box.
   As the box previously contains $\{1\}$, box = $\{1, 1, 5, 5, 10\}$ now.
   At the end of day 3, the one who pays 10 dollars get $10 - 1 = 9$ dollars prize.
   Then both 10 and 1 (either one) dollars receipts were removed, box = $\{1, 5, 5\}$ now.
   In total, the supermarket pays $2 + 1 + 9 = 12$ dollars prizes for this promotion (the output).

C.1.1 Two More Days (2 marks)

If you have understood this question, what is the answer (the cost of the promotion) if for the given explanation above, the promotion is extended by 2 more days (the 4-th and 5-th days). There is 0 (no) receipt thrown into the ballot box on the 4-th day and only receipt with value $\{2\}$ thrown into the ballot box on the 5-th day. To convince the grader that the answer (that is going to be bigger than 12 dollars) is not a random guess, continue the explanation as shown in the example above.

C.1.2 One Manual Test Case (2 marks)

If you have understood this question, what is the answer (the cost of the promotion) if you are given $d = 3$ days, there are 7 receipts in the first day $\{100, 7, 8, 70, 25, 50, 70\}$, 3 more receipts on the second day $\{120, 5, 99\}$, and no more receipt on the third (last) day. To convince the grader that the answer is not a random guess, show an explanation as shown in the example above.

C.1.3 A Loophole (7 marks)

There is a loophole in the marketing promotion which allows the following strategy: Each day, someone buys a lot from the supermarket (let's say x dollars). Instead of just throwing that x-dollars receipt into the ballot box, he/she also buys any other 1-dollar item from the supermarket and throw that additional 1-dollar receipt into the ballot box. Thus for each day, if $k > 0$, then k is even and $\frac{k}{2}$ receipts are 1-dollar worth of receipts. For 6 marks, propose an $O(n \log n)$ algorithm to solve this.

C.1.4 Solve The Full Problem (9* marks)

Suppose the supermarket realizes this loophole *before* the marketing promotion starts and revised first rule into: "A customer who wants to participate in the promotion, writes his/her personal details on the receipt that he/she paid by him/herself, and throws it into a special ballot box. *Each person can only submit one receipt per day.*". Now, propose the most efficient data structure(s) that is/are needed to solve the full problem and analyze its time complexity. To score additional 9 marks in this problem, your algorithm with the correct data structure(s) should still be in $O(n \log n)$. Note that the grading scheme of this last section is very strict: 0/1/9 for wrong answer/blank/correct answer. There is no partial marks. Thus, ensure that you answer Section C.1.3 first.

C.2 Lake Currents (20 marks)

For a boat on a large body of water, strong currents can be harnessed to help the boat reach its destination with the least amount of energy used. Your job is to help in the planning.

You are in a boat in a big lake that can be modeled as a rectangular $r \times c$ grid. The boat can move in one of the following eight directions: North, NorthEast, East, SouthEast, South, SouthWest, West, or NorthWest. At each cell, there is a current that flows in one of the 8 directions or there is no flow. If there is a current in that cell, you can choose to either go with the flow of the current, using no energy, or to move one square in any other direction, at the cost of one energy unit. If there is no current in that cell, you can move one square in any 8 directions, also at the cost of one energy unit. The boat cannot leave the boundary of the lake.

The first line of input contains two integers r and c ($1 \leq r, c \leq 1000$), the number of rows and columns in the grid. Each of the following r lines contains c characters, each a digit from 0 to 8 inclusive. The character 8 means that there is no flow, 0 means the current flows North (i.e., up in the lake grid, in decreasing row number), 1 means it flows NorthEast, 2 means East (i.e., right in the lake grid, in increasing column number), 3 means SouthEast, and so on in a clockwise manner:

```
NW 7  0  1 NE   // 0 is N(orth)
    \ | /
  W 6-8-2 E
    / | \
SW 5  4  3 SE   // 4 is S(outh)
```

Your task is to design an algorithm to go from the current location of your boat (r_1, c_1) to a target location (r_2, c_2) using the least amount of energy.

C.2.1 Manual Test Cases ($2 \times 2 = 4$ marks)

If you have understood this question, what is the answer of the following queries.

```
5 5
04125
03355
64734
72377
02062
```

0. Go from $(5, 3)$ to $(3, 4)$ (The first query has been answered for you as an example).
   There is a 0 (North)-bound flow at $(5, 3)$, go with the flow.
   Then there is a 3 (SouthEast)-bound flow at $(4, 3)$, also go with the flow.
   Then there is a 6 (West)-bound flow at $(5, 4)$, use one energy unit to go NorthEast instead.
   Then there is a 7 (NorthWest)-bound flow at $(4, 5)$, go with the flow.
   We arrive at cell $(3, 4)$ by only using one energy unit.
1. Go from $(4, 5)$ to $(1, 4)$.
2. Go from $(1, 5)$ to $(2, 2)$.

C.2.2 Special Case (7 marks)

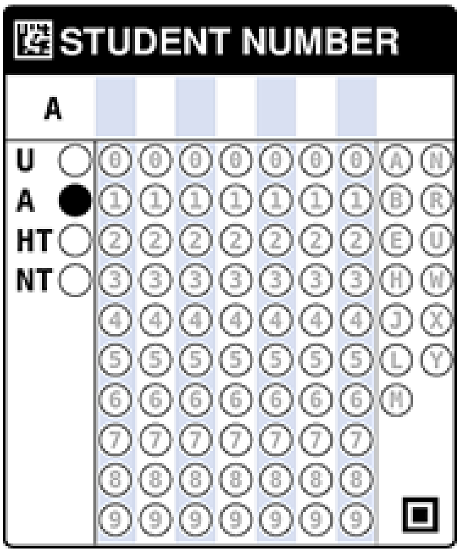
Suppose that the entire $r \times c$ grid contains only value 8, i.e., no flow at all, the lake is calm. You cannot take advantage of the flow of the current as there is none. Each movement to any neighboring direction requires 1 unit of energy. Design the best algorithm and analyze it's time complexity.

C.2.3 Solve The Problem (9* marks)

Suppose that the entire $r \times c$ grid contains random values 0..8, i.e., the currents at the lake is very chaotic. Design the best algorithm and analyze it's time complexity. Note that the grading scheme of this last section is very strict: 0/1/5/9 for wrong answer/blank/second best answer/correct answer. There is no other partial marks. Thus, ensure that you answer Section C.2.2 first.

The Answer Sheet

Write your Student Number and MCQ answers in the boxes below using **(2B) pencil**:



No	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
A															
B															
C															
D															
E															

No	16	17	18	19	20
A					
B					
C					
D					
E					

Box B.1.1. Sort These Countries

By their country names:

By their ISO 3166-1 alpha-3 codes:

Box B.1.2. What Are Your Advices?

Box B.2. The Five Modifications:

1. The `Insert(i, v)` operation – inserting value v at specific index i – now does not need to ______________________________ in $O(n)$ time before allowing v to be inserted at index i.

2. The `Search(v)` operation – previously searching for just a single index that contains value v (if any) – now has to ______________________________ ______________ when multiple copies v exists in the DLL.

3. The `Remove(i)` operation – deleting a value at specific index i – ______________________ ______________________________ to be changed.

4. The `Remove(v)` operation – previously deleting just one copy of value v from the DLL (if any) now needs ______________________________ when multiple copies v exists in the DLL.

5. The `Create(A) - Random` operation – creating a random DLL of n integers – now has to ______________________________ duplicates.

In case this sheet is detached from page 13-14, re-write your Student Number again:

STUDENT NUMBER				
A				

Box C.1.1. Two More Days

Box C.1.2. One Manual Test Case

Box C.1.3. A Loophole

Box C.1.4.* (1 if blank, 0 if wrong) Solve The Full Problem

Box C.2.1. One Manual Test Case

Box C.2.2. Special Case

Box C.2.3.* (1 if blank, 0 if wrong, 5 for second best answer) Solve The Problem

– END OF PAPER; All the Best –