ASSUMPTION UNIVERSITY

Vincent Mary School of Science and Technology
Department of Computer Science
Department of Information Technology

# Final Examination
# Semester 2/2021

| | |
|---|---|
| **Subject Code :** | CS3201/CSX3009 |
| **Subject Title :** | Algorithm Design |
| **Date :** | March 21, 2022 |
| **Time :** | **9.00 – 12.00 (3 hours)** |
| **Instructors :** | Asst. Prof. Dr. Thitipong Tanprasert (Full-Time) |

**Instructions:**

1. Read the questions carefully and answer each question completely, legibly, and concisely.
2. You must type/write your answers in your computer only. Writing answers in a paper and taking a photo using camera are not allowed.
3. An answer to a question is either a text file, a .docx file, or a Python 3 program.
4. To submit your answers, compressing (zip) all of your answers into one file and upload the zipped file to the Final Examination created as an assignment for the course. no later than 12:00 the latest. Any late submission received after 12:01 will not be graded.
5. This is an opened book examination; you can use any materials as references, including online search. However, any form of communication with anyone regarding the exam, directly or indirectly, will be considered "CHEATING".
6. You MUST turn on your camera and microphone, share your working screen (the entire screen of your PC), and record the video in MS Team for the whole examination period. The answered file will NOT BE GRADED IF THERE IS NO COMPLETELY RECORDED VIDEO CLIP.
7. This examination paper and recorded video are an intellectual property of Assumption University; you are NOT allowed to duplicate, share, or publicize it.
8. If you cheat or contribute to cheating at the exam, you will get zero score and will be considered to get the grade 'F' for this course.

---

**Marking Scale:**

| | | |
|---|---|---|
| Essay and/or Programming | 7 questions | 70 marks |

**Total 70 marks**

---

**Total 5 pages (including this page)**

**GUIDELINE:** Every question in the same page can be solved with the same algorithmic technique.

1) [10 marks] Combining message packets

In a packet switched network, a message is broken into N message packets of different lengths. Eventually they need to be combined into one message for the complete delivery. However, only two packets can be combined at a time, and the cost to combining two packets is equal to sum of their lengths.

Develop a Python program that determines the minimum possible total cost of combining these packets into one.

**INPUT:** A line containing N integers, $1 \leq N \leq 10000$, the lengths of packets

**OUTPUT:** the minimum possible total cost for combining the given N packets into one.

**EXAMPLE**

| INPUT | OUTPUT |
|---|---|
| 4 3 2 6 | 29 |

Elaboration: Combine 2 and 3 (cost = 5), then 5 and 4 (cost = 9), then 9 and 6 (cost = 15).

2) [10 marks] A 2x2 matrix multiplication function is provided. Given an integer $n$, write a Python 3 program to compute the value of Fibonacci number $F_n$ ; $10^8 \leq n \leq 10^{50}$

**INPUT:** An integer n

**OUTPUT:** The residue of the corresponding Fibonacci number divided by 2147483647

**EXAMPLE**

| INPUT | OUTPUT |
|---|---|
| 123456789 | 2053005829 |
| 12345678901234567890 | 268002575 |

HINT: According to the following fact, $F_n$ can be obtained by computing $M^{n-1}$.

$$\begin{bmatrix} F_n \\ F_{n-1} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} F_{n-1} \\ F_{n-2} \end{bmatrix}$$

$$\begin{bmatrix} F_n \\ F_{n-1} \end{bmatrix} = M \begin{bmatrix} F_{n-1} \\ F_{n-2} \end{bmatrix}$$

$$\begin{bmatrix} F_n \\ F_{n-1} \end{bmatrix} = M^2 \begin{bmatrix} F_{n-2} \\ F_{n-3} \end{bmatrix}$$

$$\begin{bmatrix} F_n \\ F_{n-1} \end{bmatrix} = M^{n-1} \begin{bmatrix} F_1 \\ F_0 \end{bmatrix}$$

3) [10 marks] A partitioning function, as that of quicksort, is provided. The partition will arrange the list A[p:r+1] into 3 parts; A[q], A[p:q], and A[q+1:r+1]. After partitioning, every value in A[p:q] is less than or equal to A[q], and every value in A[q+1:r+1] is more than A[q]. The function returns q, which is the index of the pivot.

Write a Python 3 program that utilizes the given partition function to find the k$^{th}$ order statistic, which is the k$^{th}$ smallest value in the entire list.

The program takes a sequence of numbers and a value of k as input and prints the k$^{th}$ order statistic as output.

**Hint** Upon partitioning, decide based on the returned pivot index whether to recursively search into A[p:q] or A[q+1:r+1].

**INPUT:** The first line containing N integers, $1 \leq N \leq 10000$, the lengths of packets
The last line has one integer, k.

**OUTPUT:** the k$^{th}$ order statistic of the input

**EXAMPLE**

| INPUT | OUTPUT |
|---|---|
| 2 8 3 7 4 6 5<br>4 | 5 |

Elaboration: There are three numbers that are less than 5. So 5 is the 4$^{th}$ order statistic.

4) [10 marks] The game board is an $h \times w$ rectangle, where there are $h$ rows of squares from front (first row) to back (last row) and $w$ columns of squares from left to right. Each square has a token whose value can be from 1 to 100. You want to collect as much total value of tokens as possible, subject to the following game rules:

- You start by choosing any token in the first row.
- Then, you move to a square in the next row, collect the stone on that square, and so on until you reach the last row.
- When you move from one square to a square in the next row, you can only move to the square just below it or diagonally to the left or right.

Given the values of h and w, and the value of token on each square, write a program to compute the maximum possible total token value that you can grab in one single trip from the first row to the last row.

**INPUT:** The first line has two integers; the number of rows, $h$, and the number of columns, $w$, $(1 \leq h, w \leq 9)$.
Next, there are $h$ lines of inputs. The $i^{th}$ line specifies the token values of the $i^{th}$ row from the front. Each line has $w$ integers, where each integer $m$ $(0 \leq m \leq 100)$ is the token value on that square. The integers are separated by a space.

**OUTPUT:** The maximum possible total value of tokens that you can grab.

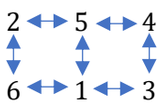
**EXAMPLE**

| INPUT | OUTPUT |
|---|---|
| 6 5<br>6 2 5 3 1<br>3 1 8 4 2<br>2 1 3 1 1<br>1 2 2 1 6<br>2 2 1 4 3<br>2 1 4 5 4 | 29 |

Elaboration: 5+8+1+6+4+5 = 29

5) [10 marks] Bubbling

Six numbers (1 – 6) are arranged as a 3x2 matrix. The only operations that can be performed on the matrix are swapping elements that are vertically adjacent or horizontally adjacent, as illustrated.




The goal is to arrange the number into the following configuration.

1 2 3
4 5 6

**INPUT:** Two lines, each line contains three numbers in the 3x2 matrix configuration

**OUTPUT:** The minimum number of swapping operations that transform the given input into the goal configuration.

**EXAMPLE**

| INPUT | OUTPUT |
|---|---|
| 1 4 5<br>2 3 6 | 4 |
| 3 2 1<br>4 5 6 | 3 |

Elaboration for 1st case: the four swaps are (4,3), (2,4), (3,5), (2,5)

6) [10 marks] This is an essay question about Bounding function technique.

a) [5 Marks] How does an appropriate bounding function improve the efficiency of a depth-first search algorithm?

b) [5 Marks] What is the essential property of a bounding function that will guarantee that the correctness of the depth-first search algorithm is maintained?

7) [10 marks] Stock investment

Today, there are $n$ stocks for Patrick to look to invest his $m$ dollars on. For stock $i$ , the current price is $c_i$ dollars, and according to Patrick's prediction, he will earn a profit of $p_i$ dollars in one year.

Find the maximum total one-year profit that Patrick can earn as the result of his stock purchase today, given that his prediction is perfectly accurate.

**INPUT:**

The first line contains 2 integers, $m$ and $n$ ($m \leq 1{,}000{,}000{,}000$ and $n \leq 200$)

Each of the following $n$ lines contain two numbers. The first number indicates the current price and the second number indicates the one-year profit from the corresponding stock.

**OUTPUT:** The maximum total profit that Patrick can earn in one year time.

**EXAMPLE**

| INPUT | OUTPUT |
|---|---|
| 5 3<br>3 1<br>3 2<br>3 1 | 2 |
| 10 4<br>4 10<br>5 4<br>6 3<br>7 11 | 14 |