Assignment Part 1 (Group, Due: 25/9/23 23:59)

Question 1:

Gale and Shapley published their paper on the Stable Matching Problem in 1962; but a version of their algorithm had already been in use for ten years by the National Resident Matching Program, for the problem of assigning medical residents to hospitals.

Basically, the situation was the following. There were m hospitals, each with a certain number of available positions for hiring residents. There were n medical students graduating in a given year, each interested in joining one of the hospitals. Each hospital had a ranking of the students in order of preference, and each student had a ranking of the hospitals in order of preference. We will assume that there were more students graduating than there were slots available in the m hospitals.

The interest, naturally, was in finding a way of assigning each student to at most one hospital, in such a way that all available positions in all hospitals were filled. (Since we are assuming a surplus of students, there would be some students who do not get assigned to any hospital.)

We say that the assignment of students to hospitals is **stable** if neither of the following situations arises.

First type of instability: There are students s and s', and a hospital h, so that:

1) s is assigned to h;
2) s' is assigned to no hospital;
3) h prefers s' to s.

Second type of instability: There are students s and s', and hospitals h and h', so that:

1) s is assigned to h;
2) s' is assigned to h' hospital;
3) h prefers s' to s;
4) s' prefers h to h'.

There is an instance: 5 students $S = \{1, 2, 3, 4, 5\}$ and two hospitals $H = \{C, M\}$, each with 2 positions available. The preference is given below. Give a stable matching and an unstable matching.

$C: < 5, 1, 2, 4, 3 >$; $M: < 5, 3, 1, 2, 4 >$;

$1: < C, M >$; $2: < C, M >$; $3: < C, M >$; $4: < M, C >$; $5: < M, C >$.

Implement the above algorithm with the provision to read an input preference text file as shown above (i.e. preference list). Output a list of stable matches. Validate your implementation with sufficient test cases.

Analyse the efficiency of your algorithm using the Big-oh notation.

Assignment Part 1 (Group, Due: 25/9/23 23:59)

Question 2:

Consider the following algorithm.

ALGORITHM Secret($A[0], A[1], \cdots, A[n-1]$)
// Input: an array $A[0], A[1], \cdots, A[n-1]$) of n numbers
// Output: an n by n array $\{B[i][j] \mid 0 \leq i, j \leq n-1\}$
1. **for** $i \leftarrow 0$ **to** $n-1$ **do**
2. **for** $j \leftarrow i$ **to** $n-1$ **do**
3. Add up array entries $A[i]$ through $A[j]$
4. Store the result in $B[i][j]$
5. **return** B

a) Suppose input $A = \{1, 2, 3, 4\}$, and each entry of B is initialized as 0, what is its output?

b) What is its basic operation?

c) How many times is the basic operation executed?

d) What is the efficiency class (Big O) of this algorithm?

Question 3:

Find the computational complexity of the following piece of code assuming that n=2m. Clearly show the detail working how complexity value is achieved.

```
for (int i = n; i > 0; i-- ) {
 for (int j = 1; j < n; j *=2) {
  for (int k = 0; k < j; k++) {
   // constant number of operations
  }
 }
}
```

Question 4:

Write a recursive algorithm to check a sentence is palindromes (ignoring blanks, lower case and upper-case differences, and punctuation marks, so that "Madam, I'm Adam" is accepted as a palindrome). Analyse the efficiency of your implementation and provide a detailed discussion of its time complexity.

Example: Please enter a sentence: Madam, I'm Adam Checking if "Madam, I'm Adam" is a palindrome: True

Assignment Part 1 (Group, Due: 25/9/23 23:59)

Question 5:

a. Explain the meaning of stability in a sorting algorithm.

b. Explain the situation why stability in sorting is desired.

c. State which algorithms are stable. Proof it with an implementation of a stable and an unstable sorting algorithm. Discuss in detail your justification. You are only required to implement 2 algorithms (i.e. 1 stable and 1 unstable).

d. Analyze the efficiency of your implementation and provide a detailed discussion of its time complexity.

   - selection sort
   - insertion sort
   - quick sort

<u>Submission Guide:</u>

Required Documentation:

- For non-programming question
  - Clearly state your answer with detail discussion of the working and analysis.
- For programming question
  - Requirements/Specification: a paragraph giving a more detailed account of what the program is supposed to do. State any assumptions or conditions on the form of input and expected output.
  - User Guide: instructions on how to compile, run and use the program.
  - Structure/Design: Outline the design of your program by describing your approach. Give a written description. Use diagrams, if required.
  - Limitations: Describe program shortfalls (if any), eg, the features asked for but not implemented, the situations it cannot handle etc.
  - Testing: describe your testing strategy (the more systematic, the better) and any errors noticed. Provide a copy of all your results of testing in this section.
  - Listings: Your source code must be included in this section.

Name your report as PxTy-ID1,ID2,ID3,ID4,ID5,ID6.

Create a folder to include all your sources and name it as PxTy-ID1,ID2,ID3,ID4,ID5,ID6. Clearly name the source code file as Q1, Q4 and Q5. Compress the folder as a .zip file before submitting.

‘x’ is your tutorial group and ‘y’ is your team number (e.g. P1T5-123411,123422,123433,123444,123455,123466 are students from tutorial P1 and team 5.)

Submit the report and the source’s zip file to separate submission tab.

Please follows closely what are stated above. Invalid format will result in mark deduction.

Assignment Part 1 (Group, Due: 25/9/23 23:59)

Presentation:

Presentation of your work is required on week 5 (26/9/23). No ppt is required. You can use your report and a demonstration of your program.

Self and Peer evaluation:

Each student needs to complete a self and peer evaluation after the submission.

If a group with 5 students received 72 for the assignment, then the following will determine the student's final score.

- Be careful when allocating percentage for each member. It should not vary too much.
- Your contribution is determined by the average contribution given by all other members and yourselves.

Student	Contribution	Final score
S1	80/100	80/120*72=48.0
S2	95/100	95/120*72=50.5
S3	95/100	95/120*72=50.5
S4	110/100	110/120*72=66.0
S5	120/100	120/120*72=72.0
	Max=120	

Assignment Part 1 (Group, Due: 25/9/23 23:59)

Marking Guidelines:

Items	Poor	Attempted	Good	Excellent	Points
Design (algorithms). For non-programming question, this will be the accuracy of the answer.	Minimal design or inaccurate deign.	Valid design with clear explanation. Minor error.	Valid design with good explanation and justification with no error.	Excellent and efficient design with clear explanation and justification with no error. Consider alternative approaches.	20
Implementation (algorithms). For non-programming question, this will be the explanation and justification.	Minimal implementation or inaccurate result.	Correct result with good programming practice (e.g. well explained and comment)	Correct result with good programming practice. Some testing to ensure correctness of the code	Correct result with good programming practice. Proof of detail testing to ensure the correctness of all cases. Efficient code with extensive effort considering all scenarios.	20
Analysis (e.g. Big-O)	Results are presented. Little analysis is made.	Some analysis and justification with minor error.	Complete analysis and justification with no error.	In depth analysis with no error. Excellent arguments and justification.	20
Documentation	Minimal documentation and didn't follow submission instruction	Basic documentation with most of the required components and follow submission instruction partially.	Good documentation with well-structured report containing all the required components. Follow submission instructions fully.	Excellent and well-structured documentation with clear and accurate content beyond the requirement (e.g. references with comparative study). Follow submission instruction fully.	20
Presentation	Partial presentation with inaccurate explanation.	Full presentation with clear and accurate explanation in most parts.	Full presentation with clear and accurate explanation for all parts.	The Presentation is exceptional. The work is consistent and professional. Complete and accurate explanation for all parts beyond the requirement.	15
Self and Peer Evaluation	Didn't submit	submitted	submitted	submitted	5