Tuesday 02 May 2023
09:30 – 11.30 BST
Duration: 1 hour 30 minutes
Additional time: 30 minutes
Timed exam – fixed start time

DEGREES OF MSc, MSci, MEng, BEng, BSc, MA and MA (Social Sciences)

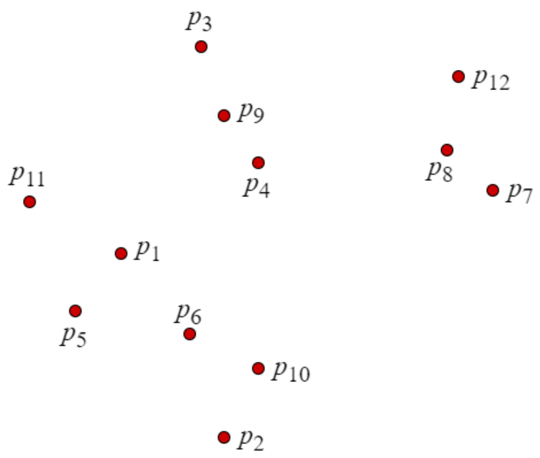
ALGORITHMICS II (H)
COMPSCI4003

(Answer all 4 questions)

This examination paper is an open book, online assessment and is worth a total of 60 marks

1. **Geometric algorithms** **[16 marks]**

(a) Consider the following set P of 12 points in the plane that are given as input to the Graham Scan algorithm:



(i) Demonstrate an execution of the Graham Scan algorithm when given the point set P as input, showing which points are added to and removed from the temporary convex hull. Your answer should be along the following lines "add p_i, add p_j, add p_k, remove p_j, …".

[6]

(ii) Give a list of points from P that belong to the final convex hull once the algorithm terminates. Ensure that the points are ordered so that the convex hull of P is formed when consecutive points in your list are joined by a line segment (and the last point is joined to the first).

[2]

(b) Let Q be a set of $n \geq 3$ distinct points in the plane that are given as input to the Graham Scan algorithm. Let q be the final point added to the convex hull during an execution of this algorithm, and let r be the pivot point.

(i) Explain why the Graham Scan algorithm does not need to check whether q needs to be excluded from the temporary convex hull.

[4]

(ii) Explain why the Graham Scan algorithm does not need to check whether r needs to be excluded from the temporary convex hull.

[4]

2. String and text algorithms **[13 marks]**

Let X and Y be two strings of lengths m and n respectively over a given alphabet Σ. Suppose that the score, denoted by T, of a highest scoring local similarity of X and Y has been computed using the Smith-Waterman algorithm, and assume that $T \geq 1$. Let S denote the dynamic programming table computed by the algorithm during its execution; thus $S(i, j)$ denotes the score of a highest-scoring local similarity obtained by substrings ending at position i of X and position j of Y.

(a) Professor Stokes wishes to compute *all* extents of *all* highest scoring local similarities of X and Y. She starts with the following code in a main method:

```
for (int i=1; i <= m; i++)
   for (int j=1; j <= n; j++)
      if (S(i, j)==T)
         recExtent(i, j, i, j);
```

Complete the body of the recursive method `recExtent` so that each extent is output along the lines of the following example: "`X: [3..10]  Y: [5..9]`". Note that the overall algorithm need not run in polynomial time.

[5]

(b) Let X=**`GCATATCG`** and Y=**`GCAGACGT`** be two strings over the alphabet {A,C,G,T}. The dynamic programming table that is computed during an execution of the Smith-Waterman algorithm is as follows:

		0	1	2	3	4	5	6	7	8
			G	C	A	G	A	C	G	T
0		0	0	0	0	0	0	0	0	0
1	G	0	1	0	0	1	0	0	1	0
2	C	0	0	2	1	0	0	1	0	0
3	A	0	0	1	3	2	1	0	0	0
4	T	0	0	0	2	2	1	0	0	1
5	A	0	0	0	1	1	3	2	1	0
6	T	0	0	0	0	0	2	2	1	2
7	C	0	0	1	0	0	1	3	2	1
8	G	0	1	0	0	1	0	2	4	3

Give an optimal alignment (with respect to similarity score) of a highest scoring local similarity of X and Y. For each position in the alignment, add the label 'm', 'd', 'i' or 's' according to whether the two characters match, a character of X must be deleted, a character of Y must be inserted, or a character of X must be substituted for a character of Y, respectively, when converting the substring of X to the substring of Y.

To *illustrate* what is expected, here is an optimal alignment of **`AACGGTCC`** and **`TAATGGCC`** with respect to similarity score, including the aforementioned labels:

```
  A A C G G T C C
| | | | | | | | |
i m m s m m d m m
| | | | | | | | |
T A A T G G   C C
```

[8]

3. Graph and matching algorithms **[15 marks]**

Suppose that an instance of the Stable Marriage problem is given, comprising a set S of students and a set L of lecturers, where $|S|=|L|=n$, for some $n \geq 2$. Consider the following version of the Gale-Shapley algorithm, called the Alternative Gale-Shapley (AGS) algorithm:

```
M = ∅ ;
assign each student and lecturer to be free ;
while (some student s is free and s has a non-empty list)
{  ℓ = first lecturer on the preference list of s ;
   // s applies to ℓ
   for (each student s′ such that ℓ prefers s to s′)
   {  delete s′ from the preference list of ℓ ;
      delete ℓ from the preference list of s′ ;
      if (s′ is assigned to ℓ in M)
         unassign s′ from ℓ in M ;  // s′ is free again
   }
   assign s to ℓ in M ;
}
output M ;
```

(a) Consider the following instance of the Stable Marriage problem in which n=4:

Students' preferences		Lecturers' preferences
s_1: ℓ_4 ℓ_1 ℓ_2 ℓ_3		ℓ_1: s_3 s_4 s_1 s_2
s_2: ℓ_3 ℓ_4 ℓ_2 ℓ_1		ℓ_2: s_4 s_1 s_3 s_2
s_3: ℓ_4 ℓ_2 ℓ_1 ℓ_3		ℓ_3: s_1 s_4 s_3 s_2
s_4: ℓ_4 ℓ_3 ℓ_1 ℓ_2		ℓ_4: s_1 s_2 s_3 s_4

Show the preference lists that remain after the AGS algorithm is applied to this instance. You need not give the matching M.

[6]

(b) Show that, for general problem instances, the AGS algorithm always produces a stable matching. You can assume that the algorithm always terminates.

[9]

4. Algorithms for hard problems **[16 marks]**

The PARTITION decision problem can be defined as follows:

PARTITION
Instance: a collection of (not necessarily distinct) positive integers $x_1, x_2, \ldots, x_n$
Question: is there a subset S of $N=\{1, 2, \ldots, n\}$ such that $\sum_{i\in S} x_i = \sum_{i\in N\setminus S} x_i$?

An instance of the JOB SHOP SCHEDULING PROBLEM (JSSP) involves a set $J=\{j_1, j_2, \ldots, j_n\}$ of n jobs, and a set of n identical machines. Each job j_i takes t_i minutes to complete. Each machine can only run for T minutes in total. Assume that each job must be processed once (and by any machine), and each machine can only process one job at a time. Moreover assume that the order in which jobs are processed by machines is not important.

A solution to JSSP is an assignment of jobs to machines such that (i) every job is processed, (ii) each machine runs for at most T minutes, and (iii) the number of machines that are used (i.e., having at least one assigned job) is minimised.

(a) By reducing from PARTITION, show that JSSP is not approximable within a factor better than 3/2 unless P=NP.

[11]

(b) Describe in outline an approximation algorithm for JSSP that has a performance guarantee of 2. You need not prove that your algorithm has a performance guarantee of 2.

[3]

(c) Does JSSP have an approximation algorithm with performance guarantee 3/2? Explain your answer briefly.

[2]