National University of Singapore

School of Computing

CS4234 - Optimisation Algorithms

(Semester 1: AY2019/20)

Tuesday, 03 December 2019, PM (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 FOURTEEN (14) printed pages, including this page.
3. This is an **Open Book Assessment**.
4. For Section A, answer **ALL** questions using the OCR form provided (use 2B pencil).
   For Section B and C, answer **ALL** questions within the **boxed space** in this booklet.
   Only if you need more space, then you can use the empty page 14.
   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 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 *and its time complexity*, e.g. run $O(n^2m)$ Dinic's algorithm on flow graph G, run $O(n^3)$ Push-Relabel on flow graph G', run $O(n^3)$ Edmonds Matching on unweighted graph G'', run $O(n^3)$ Hungarian algorithm on weighted bipartite graph G''', etc.
7. Please write your Student Number only. Do **NOT** write your name.

A	0							

This portion is for examiner's use only

Section	Maximum Marks	Your Marks	Remarks
A	34		
B	30		
C	36		
Total	100		

A MCQs (34 marks)

Select the **best unique** answer for each question.
Each correct answer worth 2 marks but each **wrong answer worth -1 mark**.
Your score for this section will not be lower than 0.

The MCQ section is not supposed to be archived to open up possibilities of reuse in the future.

B 2-Hamiltonian-Path (30 Marks)

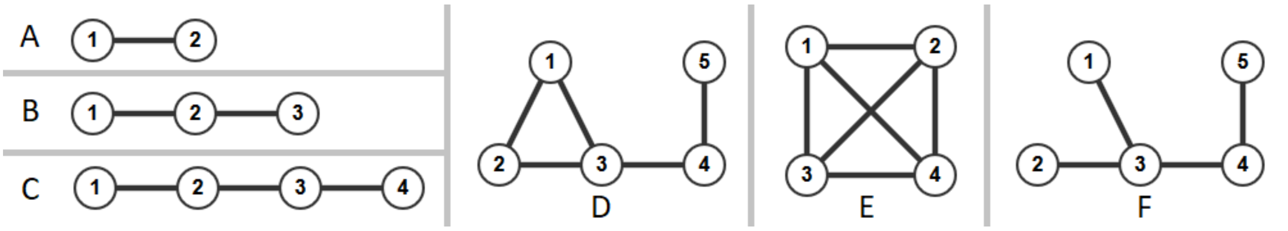
While it is not explicitly discussed in the class, we often use Hamiltonian-Path problem to prove many other problems such as Traveling-Salesman-Problem and Longest-Path problem to be NP-Hard by using reduction.

Recall the definition of a Hamiltonian-Path problem: Given a **simple and connected** graph $G = (V, E)$ with $n \geq 2$ vertices, decide whether there exists a path of vertices $v_1 \to v_2 \to v_3 \to \cdots \to v_n$ such that $(v_i, v_{i+1}) \in E$ and each vertex is visited exactly **once**. Note that Hamiltonian-Path problem is NP-Complete.

Now we define 2-Hamiltonian-Path problem as follows: Given a **simple and connected** graph $G = (V, E)$ with $n \geq 2$ vertices, decide[1] whether there exists a path of vertices $v_1 \to v_2 \to v_3 \to \cdots \to v_{2n}$ such that $(v_i, v_{i+1}) \in E$ and each vertex is visited exactly **twice**. Note that each edge can be used multiple times in the path.

Part 1 ($6 \times 1 = 6$ Marks)

Demonstrate your understanding of 2-Hamiltonian-Path problem on the following 6 graphs A-F.
For each graph, state whether it has 2-Hamiltonian-Path or not.
If it has 2-Hamiltonian-Path, then please write down one of the possible path.



My answer for graph:

A.

B.

C.

D.

E.

F.

[1]This problem is indeed a *decision* problem. For the purpose of this module title ('Optimisation Algorithms'), please treat this problem as an optimization problem: Solving the questions involving this decision problem using various techniques that we learn in class is to *optimize* your final letter grade.

Special Part (24 Marks)

We can prove that 2-Hamiltonian-Path is NP-Hard.
If you can show the non-trivial proof in under 2 hour, you will get 24 marks and can skip parts 2-5.
But beware of giving wrong proof in this special part.

If you intend to actually answer this section, please use the empty last page 14.
Your marks for this section is Part 1 + max(Special Part, Part 2+3+4+5).

Part 2 (2 Marks)

If G has a Hamiltonian-Path, does G always have 2-Hamiltonian-Path?
If yes, then explain your answer.
If no, then draw a counter-example graph and write down its Hamiltonian-Path.

Part 3 (6 Marks)

Conversely, if G has a 2-Hamiltonian-Path, does G always have Hamiltonian-Path?
If yes, then explain your answer.
If no, then draw a counter-example graph and write down its 2-Hamiltonian-Path.

Part 4 (8 Marks)

Let's take a look at a special case input where G is a **line graph**.
Is 2-HAMILTONIAN-PATH solvable in polynomial time for this case, or is it still NP-Hard?
Please explain why and give the polynomial/exponential algorithm
(with the fastest possible time complexity) to solve it.

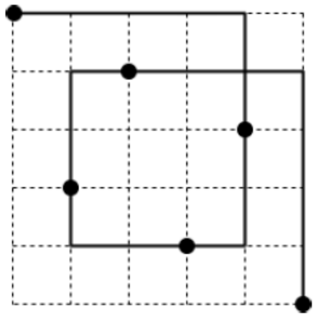
Part 5 (8 Marks)

Let's take a look at a special case input where G is a **tree**.
Is 2-HAMILTONIAN-PATH solvable in polynomial time for this case, or is it still NP-Hard?
Please explain why and give the polynomial/exponential algorithm
(with the fastest possible time complexity) to solve it.

Section B Marks = __+max(___,___+___+___+___) = ___

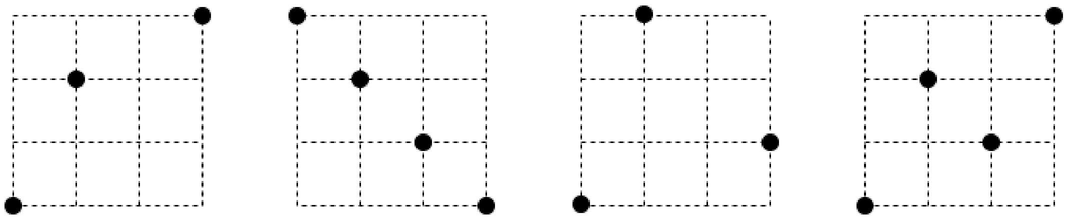
C `Minimal-Rectilinear-Path` (36 marks)

A **rectilinear path** is a path made from horizontal and vertical line segments. Given n points $(x_1, y_1), \cdots, (x_n, y_n)$ such that **no 2 points have the same x-coordinate or y-coordinate**, the MINIMAL-RECTILINEAR-PATH problem is to find the rectilinear path with the **smallest number of segments** passing through all n points. The path is allowed to intersect itself. One example is the following where $n = 6$ and the smallest number of segments is 6:



Part 1 ($4 \times 1 = 4$ Marks)

Draw the minimal rectilinear paths for the following 4 sets of points directly on the figure below.



Part 2 (2 Marks)

What is the smallest possible answer in terms of n?

Part 3 (4 Marks)

Design a linear time 2-approximation algorithm.

Better than 2-approximation

Of course, we can do much better than 2-approximation. Looking at the first 2 graphs in part 1, we notice that if the y-coordinates of the points only increase or only decrease from left to right, then they can be covered with a path of n segments[2]. Hence, if we can partition the n points into k groups with this property, then we can cover the n points using $n + k$ segments so we want to find the partition with the minimum number of such groups. If we can find this minimum, this approximation algorithm will give a ratio[3] of $1 + \frac{2}{\sqrt{n}}$.

Let's define INCREASING/DECREASING as the problem of partitioning n points, with no 2 points having same x-coordinate or y-coordinate, into the minimum number of groups such that the y-coordinates of the points in each group only increase/only decrease from left to right respectively. Define MONOTONE the same way except the y-coordinates can either increase or decrease from left to right within each group.

Part 4 (6 Marks)

Show that INCREASING can be reduced to maximum bipartite matching. Explain clearly how to construct the bipartite graph G and how to construct the partition groups from the matching.
Hint: We have done a very similar problem in tutorial.

[2] Please double check your answers for Part 1 again after reading this line.
[3] Details omitted.

Part 5 (8 Marks)

Now express INCREASING as an ILP by reducing the bipartite matching problem to maximum flow and writing that as an ILP.

Part 6 (4 Marks)

Suppose we repeat part 4 for DECREASING to get another bipartite graph G'. We can combine G and G' to form a new graph H with $v(G) + v(G')$ vertices and $e(G) + e(G')$ edges. What restrictions do we need to add to the bipartite matching of H so that it can be used to solve MONOTONE?

Part 7 (8 Marks)

Hence, express MONOTONE as an ILP by combining the bipartite matching of H with the restriction.

– End of this Paper, All the Best, You can use this Page 14 for extra writing space –

Section C Marks = (__+__+__)+__+(__+__)+__ = ___