Family Name: . . . .
Other Names: . . . .

Signature: . . . .
ID Number: . . . .

COMP 261 Test 1

22 April 2021

Instructions

- Time allowed: **50 minutes** .
- Answer **all** the questions. There are 40 marks in total.
- Write down your answers on the blank pages/your own white papers and hand in all sheets.
- If you think some question is unclear, ask for clarification.
- This test contributes 25% of your final grade.
- You may use paper translation dictionaries, and non-programmable calculators.
- You may write notes and working on this paper, but make sure your answers are clear.

Questions	**Marks**	
1. Graph	[30]	
2. Fourier Transform	[10]	
	TOTAL:	

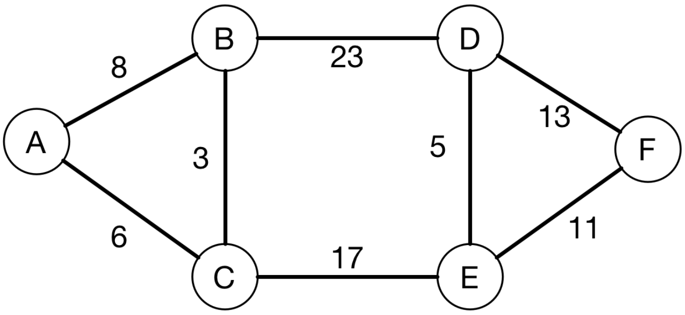
Student ID: . . . .

SPARE PAGE FOR EXTRA ANSWERS

Cross out rough working that you do not want marked.
Specify the question number for work that you do want marked.

Question 1. Graphs [30 marks]

Consider the following graph, all the edges are undirected.



(a) [8 marks] Write the *outgoing* adjacency list of each node.

(b) [8 marks] Show how to use *Dijkstra's algorithm with early stop* to search for the shortest path from node A to node F. You should show (1) at each step, the *elements in the fringe* and the *element to be visited next*, and (2) the *final shortest path* as a sequence of nodes.

- Tip: each element is represented in the format of $\langle node, prevNode, costSoFar \rangle$.

(c) [9 marks] In the articulation points algorithm, the depth-first search follows the nodes alphabetically, and get depth(A)=0, depth(B)=1, depth(C) = 2, depth(E) = 3, depth(D) = 4, depth(F) = 5. Calculate the *reach back* of each node.

(d) [5 marks] Show how to use the Prim's algorithm to find the *minimum spanning tree* of the graph. You should show the edges and their weights (e.g., AC 6) in the order of being added into the tree by the Prim's algorithm.

Question 2. Fast Fourier Transform [10 marks]

Below is the partial code of the FFT. The input is an array of double, and the output is an array of complex numbers.

```
public ComplexNumber[] FFT(double[] x) {
  int N = x.length;
  if (N == 1) {
    // TODO 1: termination criteria
  }

  ComplexNumber[] W = ComplexNumber[N];
  for (int k = 0; k < N; k++) {
    W[k] = ComplexNumber.exp(new ComplexNumber(0, -2*pi*k/N));
  }

  double[] xeven = double[N/2];
  double[] xodd = double[N/2];

  for (int k = 0; k < N/2; k++) {
    xeven[k] = x[k*2];
    xodd[k] = x[k*2+1];
  }

  ComplexNumber[] Xeven = FFT(xeven);
  ComplexNumber[] Xodd = FFT(xodd);

  ComplexNumber[] X = ComplexNumber[N];
  // TODO 2: calculate X from Xeven, Xodd and W[k]

  return X;
}
```

(a) [8 marks] Complete the two TODO items in the partial code.

(b) [2 marks] The code may not work for some input arrays. For which input arrays the code will not work?

* * * * * * * * * * * * * * *