

CITS2200 Project 2021

You may work individually or in pairs on this project.

Problem Specification

You must implement the following four functions, and provide a report on your implementations. Each function is worth a different number of marks for a total of 40 marks. Details for what the mark breakdowns mean are provided below.

All questions consider a theoretical model of a computer network. This network contains N distinct devices, each of which is assigned a unique integer id in the range $[0, N)$. A pair of devices in the network can be physically linked such that one device is able to transmit network packets to the other. It is not necessarily the case that a physical link can be used to transmit in both directions, but some links may be. If a pair of devices are not physically linked, they are still able to transmit network packets if there is a sequence of physical links from the sending device to the receiving device such that the packet can be passed along via multiple devices to reach its destination. The number of physical links a packet passes through to reach its destination is called the number of hops it takes or the distance it travels.

For each of the below functions, a specification of the structure of the network is passed as an `int[][]` argument called `adjlist`. A device with id `v` is considered adjacent to a device with id `u` if they are physically linked such that `u` is able to transmit a packet directly to `v`. The array `adjlist[u]` is a list of all the device ids that are adjacent to `u`. For example, if `adjlist[1]` is the array `{0, 2}`, this means that device 1 is able to transmit packets to devices 0 and 2.

Time complexity specification in this document use D for the number of devices in the network, L for the number of links, and N for the size of the network ($D + L$).

`public boolean allDevicesConnected(int[][] adjlist)` (6 marks)

Determine if all of the devices in the network are connected to the network. Devices are considered to be connected to the network if they can transmit (including via other devices) to every other device in the network. If all devices in the network are connected, then return `true`, and return `false` otherwise.

Arguments:

- `int[][] adjlist` : The structure of the network

Return: `boolean` indicating whether all of the devices in the network are connected to the network

Marks (6 total):

- Correctness: +3 marks
- Complexity:
 - $O(N)$: +1 mark
- Quality: +2 marks

public int numPaths(int[][] adjlist, int src, int dst)
(10 marks)

Determine the number of different paths a packet can take in the network to get from a transmitting device to a receiving device. A device will only transmit a packet to a device that is closer to the destination, where the distance to the destination is the minimum number of hops between a device and the destination.

Arguments:

- `int[][] adjlist` : The structure of the network
- `int src` : The device id of the transmitting device
- `int dst` : The device id of the receiving device

Return: `int` The number of possible different paths in the network that a packet may take from the transmitting to receiving device

Marks (10 total):

- Correctness: +4 marks
- Complexity:
 - $O(N^2)$: +1 mark
 - $O(N)$: +2 marks
- Quality: +3 marks

**public int[] closestInSubnet(int[][] adjlist, short[][]
addrs, int src, short[][] queries)** (10 marks)

Compute the minimum number of hops required to reach a device in each subnet query. Each device has an associated IP address. An IP address is here represented as an array of exactly four integers between 0 and 255 inclusive (for example, `{192, 168, 1, 1}`). Each query specifies a subnet address. A subnet address is specified as an array of up to four integers between 0 and 255. An IP address is considered to be in a subnet if the subnet address is a prefix of the IP address (for example, `{192, 168, 1, 1}` is in subnet `{192, 168}` but not in `{192, 168, 2}`). For each query, compute the minimum number of hops required to reach some device in the specified subnet. If no device in that subnet is reachable, return

`Integer.MAX_VALUE`.

Arguments:

- `int[][] adjlist` : The structure of the network
- `short[][] addr` : An array of IP addresses such that device id `i` has address `addr[i]`
- `int src` : The device id of the transmitting device
- `short[][] queries` : An array of queries where each query is a subnet prefix

Return: `int[]` of number of hops required to reach each subnet from `src` .

Marks (10 total):

- Correctness: +4 marks
- Complexity: (where `q` is the number of queries)
 - $O((N + Q) \lg N)$: +2 marks
 - $O(N + Q)$: +1 mark
- Quality: +3 marks

`public int maxDownloadSpeed(int[][] adjlist, int[][] speeds, int src, int dst)` (14 marks)

Compute the maximum possible download speed from a transmitting device to a receiving device. The download may travel through more than one path simultaneously, and you can assume that there is no other traffic in the network. If the transmitting and receiving devices are the same, then you should return `-1` .

Each link in the network has a related speed at the same index in the `speeds` array (e.g. the link described at `adjlist[0][1]` has its related speed at `speeds[0][1]`). Speeds may be asymmetric (that is - the speed in one direction of a link may be different to the speed in the other direction of a link).

Arguments:

- `int[][] adjlist` : The structure of the network
- `int[][] speeds` : The maximum speed of each link in the network
- `int src` : The device id of the transmitting device
- `int dst` : The device id of the receiving device

Return: `int` The maximum download speed from the transmitting device to the receiving device

Marks (14 total):

- Correctness: +5 marks
- Complexity:
 - $O(D \cdot L^2)$: +3 marks

- $O(D^2 L)$: +1 mark
- $O(D^3)$: +1 mark
- Better is possible, but will not receive any additional marks
- Quality: +4 marks

Required Work

You must write a public class called `MyProject` that implements the provided `Project` interface (see `Project.java`)

- Your class must have a zero-argument constructor, which will be the only one used when marking
- Your code must be in a file named `MyProject.java`
- `MyProject.java` must include your full name(s) and student number(s) as a comment as the first line of the file
 - For example: `// Ada Lovelace (21234567), Margaret Hamilton (20123456)`
- **DO NOT** modify `Project.java` in any way
- You may use anything from the Java standard library
- You may **NOT** use anything from the CITS2200 lab package or any other sources

You must write a report explaining your implementation

- Your report must be provided as a PDF named `report.pdf`
- Your report must for each problem:
 - Explain why your chosen algorithm will give the correct answer (that is, a logical argument for why it is correct)
 - Provide an analysis explaining the time complexity of your implementation (and memory complexity if relevant)

Testing

Your code must compile correctly when compiled with:

```
javac Project.java MyProject.java SampleProjectUnitTest.java
```

After compiling, you can test your code using the provided sample unit tests with:

```
java SampleProjectUnitTest
```

You are encouraged to copy `SampleProjectUnitTest.java` and extend it to add your own tests.

Assessment

Each problem will be marked separately and the results combined for a total of 40 marks (mark distributions provided above). For each problem, you will receive marks based on:

- Correctness
 - Compiles without error
 - Gives correct results
- Complexity
 - Efficient code
 - Complexity targets are provided above
- Quality
 - Convincing and well presented arguments in report
 - Code readable and easy to follow, including sufficient commenting

For all components, you will be assessed on how well the evidence demonstrates your understanding of the content.

- Your submitted work will serve as evidence of understanding of the project content
- Any work that is not your own original work does not constitute evidence of understanding
- You are permitted to research prior work that others have done on these problems, but any work you reference must be cited in your report
- Your code must be wholly your own original work
 - Familiarise yourself with the UWA academic misconduct policy
 - Copying, transcribing, or otherwise reproducing the work of others without demonstrating an understanding of the logic therein is academic misconduct
 - UWA has serious consequences based on the severity of academic misconduct ranging from marking penalties to expulsion

Submission

Submit only `MyProject.java` and `report.pdf` via [Moodle](#).