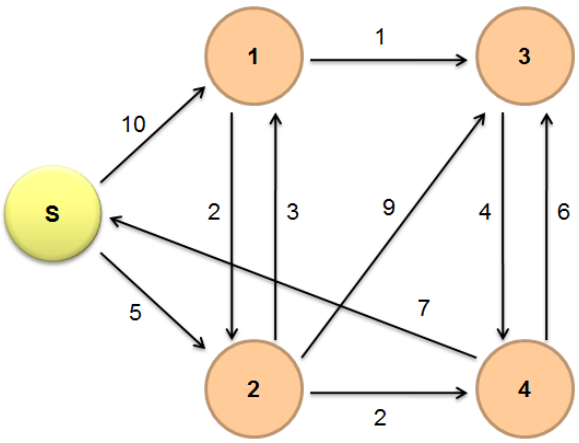
# COMP9313 2017s2 Project 2

## Single-source shortest path

Given a graph and a node "s", find the shortest distances together with the paths from "s" to all nodes. For example, the shortest distance from s to 3 is 9 with path s->2->1->3.




### Input files:

In the input file, each line is in format of:

"EdgeId FromNodeId ToNodeId Distance".

In the above example, the input contains:

```
0 0 1 10.0
1 0 2 5.0
2 1 2 2.0
3 1 3 1.0
4 2 1 3.0
5 2 3 9.0
6 2 4 2.0
7 3 4 4.0
8 4 0 7.0
9 4 3 6.0
```

This sample file "tiny-graph.txt" can be downloaded at:

https://webcms3.cse.unsw.edu.au/COMP9313/17s2/resources/12283

### Output:

Set the number of reducers to 1. The single output file contains distances of the given node to all nodes. Each line is in format of "TargetNodeID\tDistance\tPath". The distances are of double precision, and each path is a sequence of nodes on the shortest path. Remove the nodes that are not reachable to the query node, and sort the output by TargetNodeID according to its numeric value. Given the example graph, the output file is like:

```
0\t0.0\t0
1\t8.0\t0->2->1
2\t5.0\t0->2
3\t9.0\t0->2->1->3
4\t7.0\t0->2->4
```

### Code format:

Name your java file as "SingleSourceSP.java", and put it in the package "comp9313.ass2". Your program should take three parameters: the input folder containing the graph file, the output folder storing the final result file, and the query source node ID.

## Hints:

1. One difficulty of this problem is how to do the iterative MapReduce jobs. You can download the code template at: https://webcms3.cse.unsw.edu.au/COMP9313/17s2/resources/12284, which may help you solve this problem.
2. Another difficulty is how to check the termination criterion, and you are required to utilize the counter to do this job.
3. After the iterations are finished (i.e., all shortest distances are found), the output format is not as required. Thus, you need to use another round map/reduce to convert the output to the desired format. Please use the output folder given in the parameter to store the final result.

## Documentation and code readability

Your source code will be inspected and marked based on readability and ease of understanding. The documentation (comments of the codes) in your source code is also important. Below is an indicative marking scheme:

| |
|---|
| Result correctness: 80% |
| Code structure, Readability, and Documentation: 20% |

## Submission:

Deadline: Sunday 17th Sep 09:59:59 PM

Log in any CSE server (williams or wagner), and use the give command below to submit your solutions:

$ give cs9313 assignment2 SingleSourceSP.java

Or you can submit through:
https://cgi.cse.unsw.edu.au/~give/Student/give.php

If you submit your assignment more than once, the last submission will replace the previous one. To prove successful submission, please take a screenshot as assignment submission instructions show and keep it by yourself.

## Late submission penalty

10% reduction of your marks for the 1st day, 30% reduction/day for the following days.

## Plagiarism:

The work you submit must be your own work. Submission of work partially or completely derived from any other person or jointly written with any other person is not permitted. The penalties for such an offence may include negative marks, automatic failure of the course and possibly other academic discipline. Assignment submissions will be examined manually.

Relevant scholarship authorities will be informed if students holding scholarships are involved in an incident of plagiarism or other misconduct.

Do not provide or show your assignment work to any other person - apart from the teaching staff of this subject. If you knowingly provide or show your assignment work to another person for any reason, and work derived from it is submitted you may be penalized, even if the work was submitted without your knowledge or consent.