Exercises:

Problem 1: (45 pts)

Suppose that there are two types of jobs, say J1-type and J2-type, and two models of machines, say M1-model and M2-model. The processing time for each type of job on each model of machine is shown by Table 1.

	Machine	
Job	M1	M2
J1	8	12
J2	10	18

Table 1: Processing time in minutes.

Suppose that we have 5 jobs, 2 in J1-type and 3 in J2-type, to be processed by 5 machines, 3 of M1-model and 2 of M2-model.

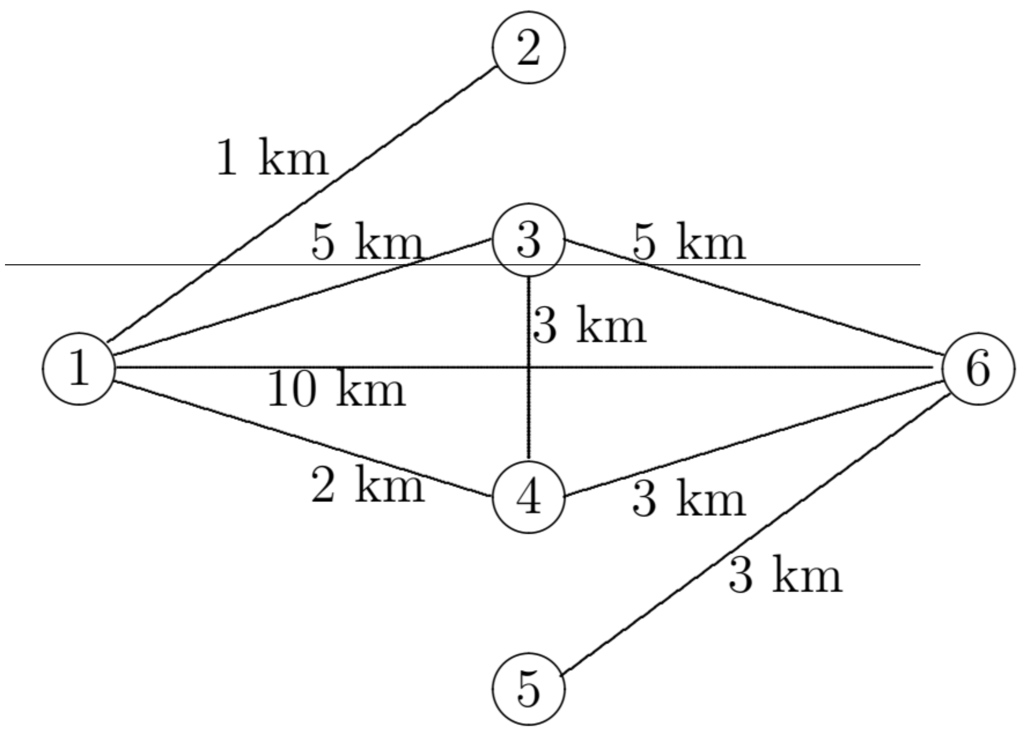
A safety rule says that each job can be assigned to exactly one machine and each machine can handle exactly one job. A profit rule says that shorter processing time costs less. As the shop manager, you are asked to have a most cost-effective assignment plan that abides by the safety rules.

(a) (10 pts) Formulate the problem as a weighted matching problem. Be sure to write in explicit form.
(b) (10 pts) Derive its dual problem in explicit form.
(c) (15 pts) Starting from the zero matching, use an algorithm you learned in the class to find an optimal assignment plan, step by step.
(d) (10 pts) Find its dual optimal solution, step by step. What's the value of this dual information?

Problem 2: (45 pts)

Given a road map as shown below, find a closed tour with the shortest distance for the Chinese postman to travel through all streets at least one time starting from node 1.

Please be sure to show your work step by step.



(a) (10 pts) Write an explicit non-bipartite matching model to help the postman.
(b) (5 pts) Derive an explicit dual problem of the matching model.
(c) (5 pts) What are the optimality conditions associated with the primal-dual pair?
(d) (20 pts) Use a matching algorithm you learned in 766 to find an optimal matching solution of (a), STEP by STEP.
(e) (5 pts) Generate an optimal tour for the postman to travel.

Problem 3: (10 pts)

For a given maze shown in the figure below, we are looking for an efficient travel path from *S* to *T*. Please model this maze problem as a network flow problem and use the theory (algorithm) you learned in OR/ISE 766 to find such a path.

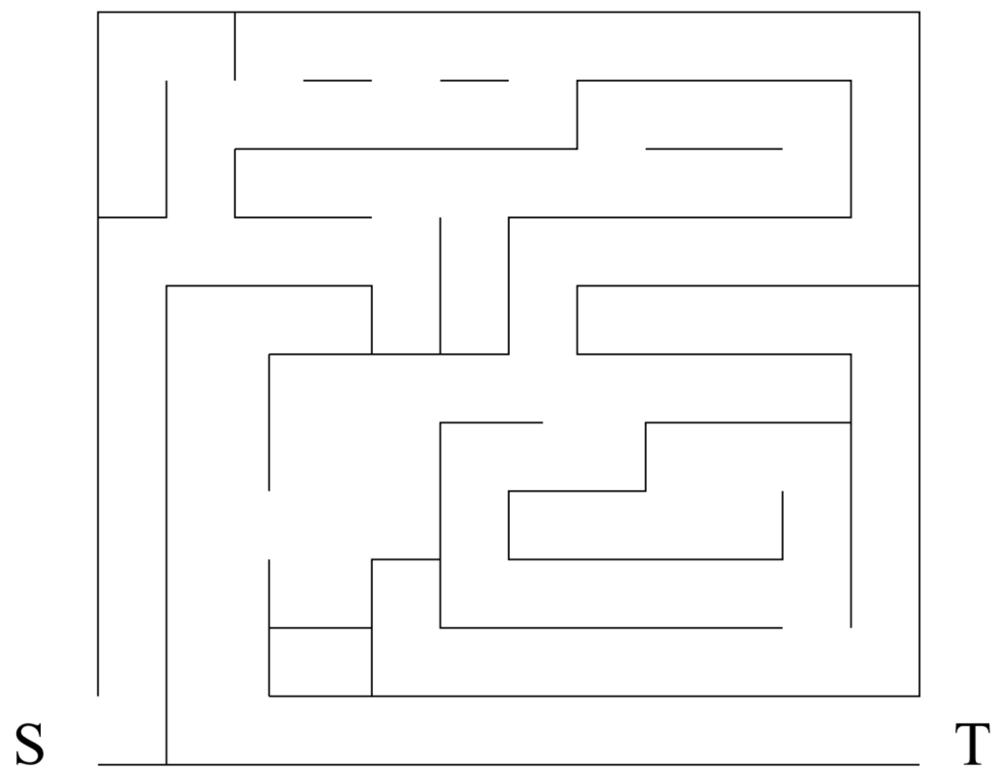


Figure: Maze Problem