

# Communications Network Design

## Class Exercise 4: before lecture, Wednesday June 3rd.

Matthew Roughan  
<matthew.roughan@adelaide.edu.au>

1. Given the network costs, and traffic shown in Figure 1, apply the following algorithms. Show all working!

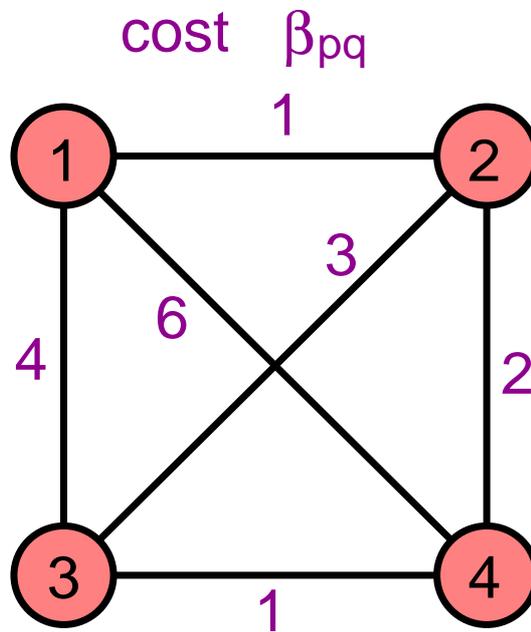


Figure 1: A network.

- (a) Apply Kruskal's algorithm. Explain what the solution represents.
- (b) Apply Prim's algorithm. Explain what the solution represents.
- (c) Compare and contrast the results of each algorithm.

2. Write a matlab program to implement Prim's algorithm. The task is similar to writing your earlier Dijkstra code. Follow the following instructions carefully.

(a) The solution should be in the form of a function, in a ".m" file, with the name of the file being:

`prim_N.m`

where N is your student number.

(b) The function should be called as follows:

`L = prim_N(beta);`

where the inputs `beta` = the cost matrix, i.e.

$beta_{i,j}$  = the cost of link  $i \rightarrow j$ .

Note that for this problem we will assume links are undirected, i.e.,  $\beta_{i,j} = \beta_{j,i}$ .

The output  $L$  is a list of links in the form of an  $(M - 1) \times 2$  array, where  $M$  is the number of nodes in the network ( $M - 1$  is the number of links in a spanning tree). Each row of  $L$  should have the two nodes connected by the link (from the spanning tree).

Please make sure that your array  $L$  is sorted so that  $L(i, 1) < L(i, 2)$  for all  $i$ , and so that  $L(i, 1) \leq L(i + 1, 1)$ , and if  $L(i, 1) = L(i + 1, 1)$  then  $L(i, 2) < L(i + 1, 2)$ . The `sort` function may be useful in this regard.

- (c) Your exercise will be checked by automatically in the same fashion as the problem on Dijkstra, but there will be no test examples – you must test the program yourself.
- (d) Your program should check for common error conditions.
- (e) Your code must be standalone, in one file! No subroutines!
- (f) Your code should produce NO outputs other than those requested. It should not print out any values. Please be careful about this. The scripts are automatically marked, and any extra outputs will cause the script to crash, and you will receive zero.
- (g) **Handin instructions:** We will use MyUni for handing up the code part of assignments. You need to make sure your matlab file has the right name, and then
- i. Go to Communications Network Design on the MyUni webpage.
  - ii. Select Assignments and then Assignment 4
  - iii. Select the HANDIN link.
  - iv. Enter a comment. MyUni requires one, but we don't read them, so it can be almost anything.
  - v. Attach the assignment file
  - vi. Click 'Submit' - NOT Save.

**Note that you can only submit your assignment once. Make sure that you have done all that is required before continuing.**

Please submit the written part of your assignments at your lecture, but use the above instructions to submit your code.