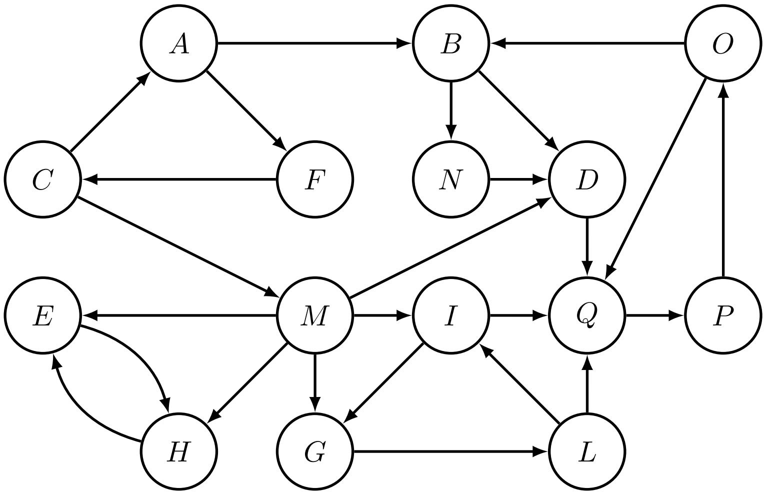
5CCS2FC2: Foundations of Computing II

Tutorial Sheet 5

5.1 Consider the following graph $G = (V, E)$ describing a set which tasks that must be completed before or concurrently with which others:



(i) Use the algorithm described in class to identify which tasks must be performed concurrently.

(ii) Draw the component graph G^{SCC}.

(iii) Perform a topological sort on the component grap G^{SCC} to identify an order in which the task can be scheduled?

(iv) How many ways can the number of number of strongly connected components change if a single additional edge is added to G?

5.2 Consider the following naïve algorithm for the minimum spanning tree problem

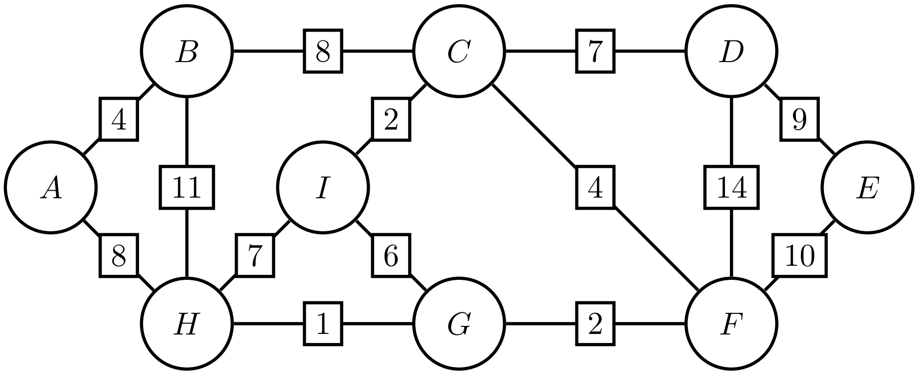
Step 1) If the graph contains at most one edge, return this as the minimum spanning tree.

Step 2) Otherwise, divide the graph G into two subgraphs G_1 and G_2 that differ in size by at most 1 vertex.

Step 3) Identify a minimum spanning tree T_1 for G_1 and a minimum spanning tree T_2 for G_2. (Do this by recursively applying this algorithm to each subgraph)

Step 4) Connect the two spanning trees T_1 and T_2 with the lightest edge between then.

(i) Apply this algorithm to the following graph:



(ii) Does the algorithm always return a spanning tree?

(ii) Does the algorithm always return a *minimum* spanning tree?