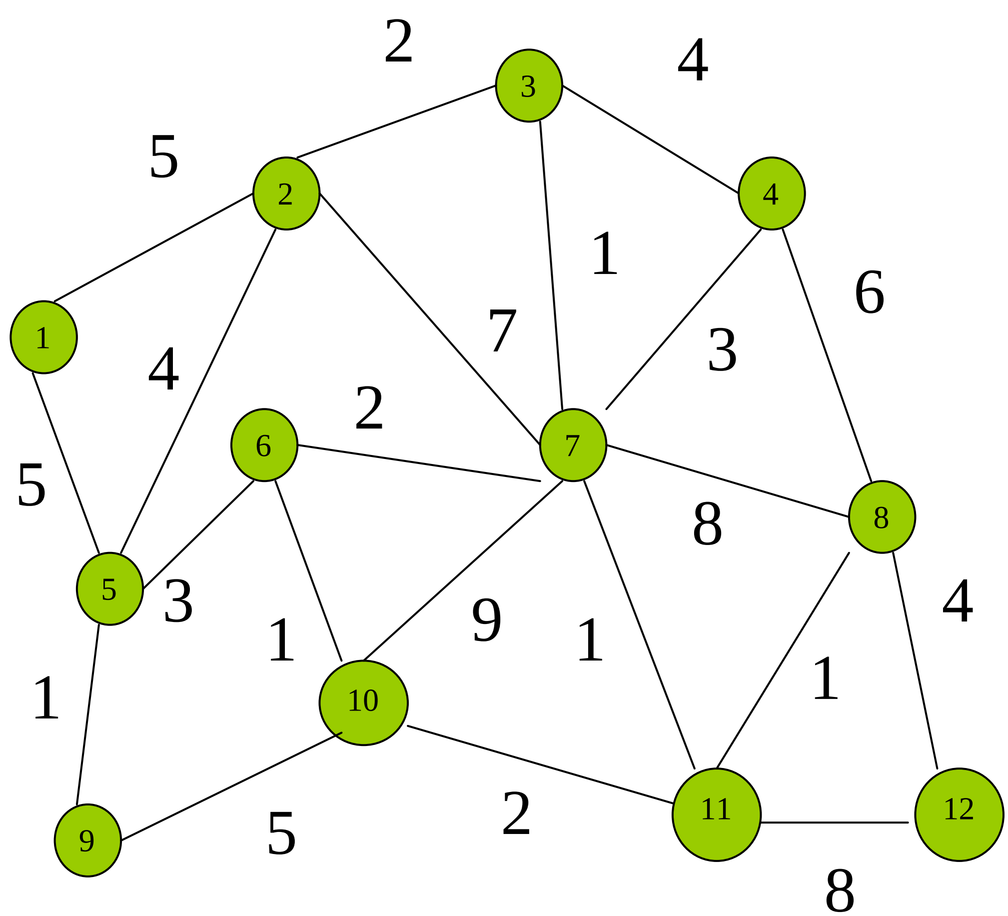
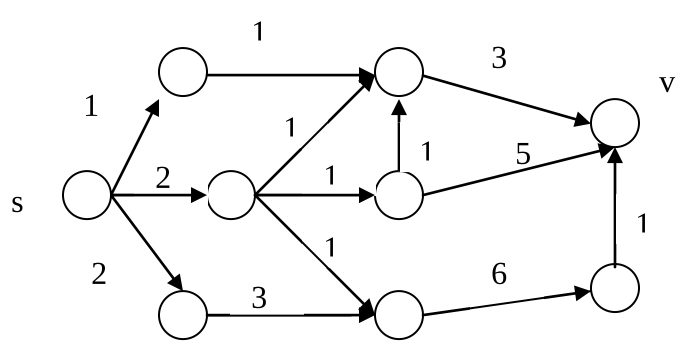
CS4335 Assignment 1 (Due Week 6, Friday, Oct. 7, 2016)

Question 1: (40 points) Consider the following graph.



Use Prim's algorithm to compute a minimum spanning tree.

Question 2: (60 points) Find the shortest path from *s* to *v* in the following graph using Dijkstra's algorithm. (Intermediate steps are required. The backtracking for finding the shortest path is required.) (Lecture 4)



Bonus: (0 Marks in the final exam since the solution can be found on internet.) This is a very good exercise since it is much harder than the next question.

The wildly popular Spanish-language search engine El Google needs to do a serious amount of computation every time it recompiles its index. Fortunately, the company has at its disposal a single large supercomputer, together with an essentially unlimited supply of high-end PCs.

They've broken the overall computation into n distinct jobs, labeled $J_1, J_2, \ldots, J_n$, which can be performed completely independently of one another. Each job consists of two stages: first it needs to be preprocessed on the supercomputer, and then it needs to be finished on one of the PCs. Let's say that job J_i needs p_i seconds of time on the supercomputer, followed by f_i seconds of time on a PC.

Since there are at least n PCs available on the premises, the finishing of the jobs can be performed fully in parallel—all the jobs can be processed at the same time. However, the supercomputer can only work on a single job at a time, so the system managers need to work out an order in which to feed the jobs to the supercomputer. As soon as the first job in order is done on the supercomputer, it can be handed off to a PC for finishing; at that point in time second job can be fed to the supercomputer; when the second job is done on the supercomputer, it can proceed to a PC regardless of whether or not the first job is done (since the PCs work in parallel); and so on.

Let's say that a schedule is an ordering of the jobs for the supercomputer, and the completion time of the schedule is the earliest time at which all jobs will have finished processing on the PCs. This is an important quantity to minimize, since it determines how rapidly El Google can generate a new index.

Give a polynomial-time algorithm that finds a schedule with as small a completion time as possible.

Bonus: (1 Mark in the final exam) Keyboard Design.

Let us consider a keyboard containing one row and *n* characters. Let $c_1, c_2, \ldots, c_n$ be the frequency of the *n* characters. A mechanical hand is always at the left side of the row after typing. The total distance moved by the mechanical hand is defined as

$$\sum_{i=1}^{n} 2i \times c_i$$

Design the layout of the row containing the *n* characters so that the total distance moved by the mechanical hand is minimized. Prove that your algorithm is correct.