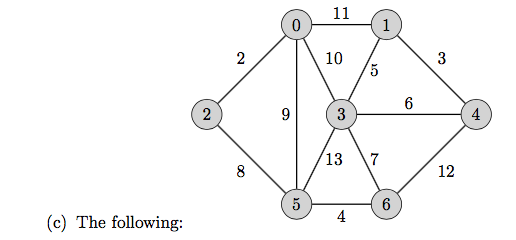
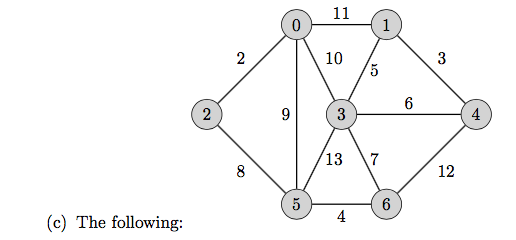
Complete hw by giving answers to the questions below, then upload the document in Canvas:

Exercise 6.3 Execute the Prim-Jarník algorithm by hand on the following graphs. Show your work.



Exercise 6.6 Execute Dijkstra’s algorithm on each of the graphs in question R-6.3, starting from s = 3.   
(c)



Exercise 6.9 For each of the following problems: design a greedy algorithm that solves the problem; describe your algorithm with clear pseudocode; and prove the time efficiency class of your algorithm.

(b) wiggle sort

input: a list L of n distinct comparable elements

output: a list S containing the elements of L in wiggle order

A sequence S = ⟨s0,s1,...⟩ is in wiggle order when the elements alternate between a less-than and greater-than relationship, so

s0 < s1 > s2 < s3 > s4 <....

(c) 2-coloring  
input: an undirected graph G = (V, E)  
output: a sequence coloring = { (v, c) |v𝜖V and c𝜖{1, 2} such that each v𝜖V appears exactly once in coloring, and no adjacent vertices are assigned the same color }

(e) Recall that a graph G = (V, E) is defined by a set of vertices V and set of edges E. An  
edge e𝜖E is incident to vertex v𝜖V when v is one of e’s endpoint vertices.

isolated vertex

input: a graph G with n vertices and m edges  
output: True if G contains a vertex with no incident edges, or False otherwise

The two most common data structures for graphs are the adjacency lists and adjacency matrix. You can choose whether the input to your algorithm is in adjacency lists or adjacency matrix format.  
Hint: A decent algorithm for this problem runs in O(n2) time, and an optimal one takes O(n) time.

Answer:

(b)

(c)

(e)

Exercise 7.1 Compute the power set of {a, b, c, d} using the algorithm from Subsection 7.5.4. Show your work.

Solution:

Exercise 7.5 For each of the following problems: design an exhaustive search or optimization algorithm that solves the problem; describe your algorithm with clear pseudocode; and prove the time efficiency class of your algorithm. When writing your pseudocode, you can assume that your audience is familiar with the candidate generation algorithms in this chapter, so you can make statements like “for each subset X of S” without explaining the details of how to generate subsets.

(a) subset sum problem

input: vector X of n distinct integers, and a target integer k

Output: a subset of X whose sum of elements is exactly k

(c) Pythagorean triple problem

input: two positive integers a, b with a < b

output: a Pythagorean triple (x,y,z) such that x,y and z are positive integers, a ≤ x ≤ y ≤ z ≤ b, and x2 + y2 = z2, or None if no such triple exists

Answer:

(a)

(c)