Final Exam Section 1 Spring 2017, Part B Key
CS146: Data Structures and Algorithms
Instructor: Katerina Potika

Question

Name the three techniques we presented in our class. Give an example of an algorithm that are based on each technique and what problem they solve. [2pts/each entry: total 18pts]

Name of Technique	Algorithm	Problem

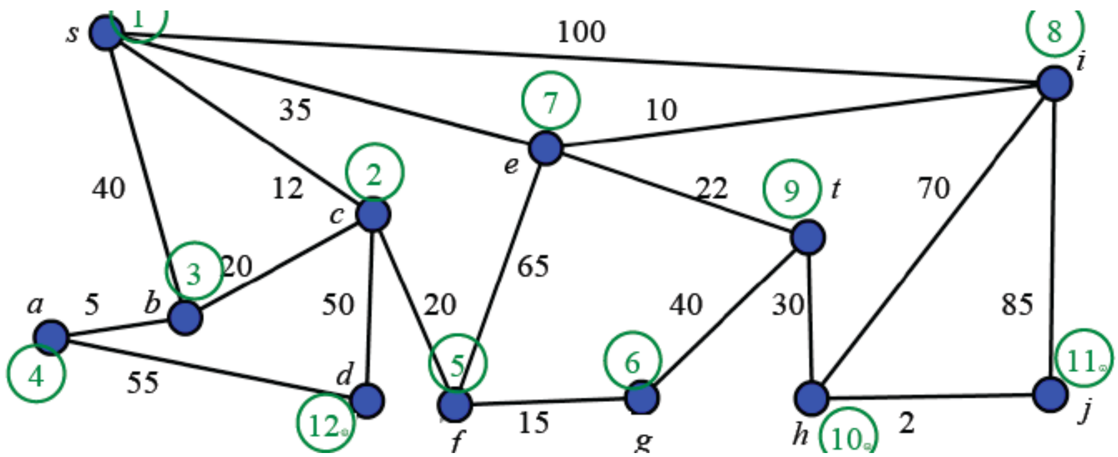
Take any two and describe one similarity and one difference. [4pts]

Question 2

a. (14pts) You have a startup with several offices as shown in the next graph; you want to lease phone lines to connect them up with each other. The phone company charges different amounts of money to connect different pairs of offices (weight on edges). You want a set of lines that connects all your offices with minimum total cost.

I. (2pts) What are you trying to compute? *MST*
II. (2pts) Name the algorithm you will use. *Prim or Kruskal*
III. (10pts) Give the order in which you select edges (endpoints) inside the next table, the total money (cost) you have to spend. If the algorithm you use needs a starting vertex, start it on vertex s.

Solution:



Q:	**vertex**	**keys**	
	s	0	√
	a	~~∞~~ ~~∞~~ 5	√
	b	~~∞~~ ~~40~~ 20	√
	c	~~∞~~ 12	√
	d	~~∞~~ ~~∞~~ 50	√
	e	~~∞~~ 35	√
	f	~~∞~~ ~~∞~~ 20	√
	g	~~∞~~ ~~∞~~ 15	√
	h	~~∞~~ ~~∞~~ ~~70~~ 30	√
	i	~~∞~~ ~~100~~ 10	√
	j	~~∞~~ ~~∞~~ ~~85~~ 2	√
	t	~~∞~~ ~~∞~~ ~~40~~ 22	√

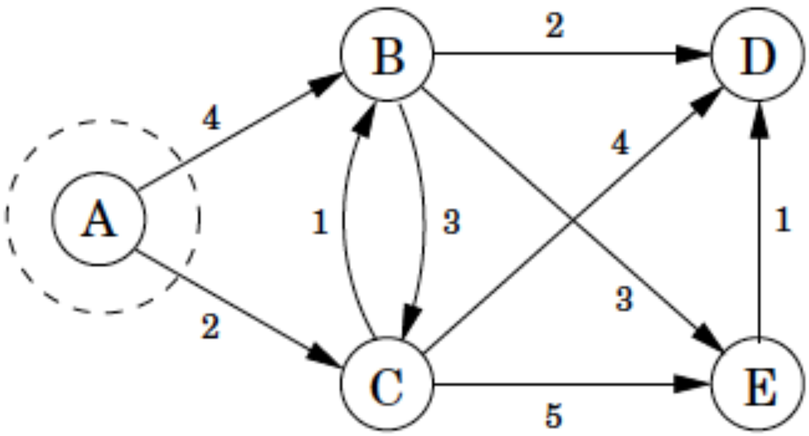
Question 3- Pathfinder-edis

(14pts) You are designing a video game in your start up, where you have cities and roads connecting them (see given graph, weight is the distance between cities). You want to use a "Pathfinding" method to plot route from city A to all other cities.

a. (2pts) What problem are you trying to solve? *Single Source Shortest Path*

b. (2pts) What is a good algorithm? *Dijkstra*

c. (10pts) Fill the next table to find the routes



Solution:

Set	A	B	C	D	E
A	0	∞	∞	∞	∞
C	-	4	2	∞	∞
B		3	-	6	7
D		-		5	6
				-	6

Question 4 – Mars here we come

(20 pts) You are part of Mars One that aims to establish a permanent human settlement on Mars. Several unmanned missions will be completed, establishing a habitable settlement before carefully selected and trained crews will depart to Mars. In one of them you have a spaceship of capacity W and some items with weight and possible benefit to the settlement. You have to decide which items fit in the spaceship and maximize the total benefit. The items have this nice property that the order of the items when sorted by increasing weight is the same as their order when sorted by decreasing value. Give an efficient algorithm to find an optimal solution and argue that your algorithm is optimal. Name the technique you used and compute the running time of your algorithm?

Let $i_1,\ i_{2,}\ \dots, i_n$ be the items with benefit $b_1, b_2, \dots, b_n$ and weight $w_1, w_2, \dots, w_n$.

Fact: $w_1 \le w_2 \le .. \le w_n$ and $b_1 \ge b_2 \ge \dots \ge b_n$

Solution:

$$
\begin{array}{ll}
w = 0 & \text{// knapsack weight} \\
S = \emptyset & \text{// knapsack content} \\
\quad \textbf{for } (i = 1;\ i \le n;\ i{+}{+}) & \\
\qquad\qquad \textbf{if } (w + w_i \le W) & \\
\qquad\qquad\qquad w \mathrel{+}= w_i & \\
\qquad\qquad\qquad S = S \cup \{i\} &
\end{array}
$$

The following linear-time algorithm does the job. This algorithm solves this special Knapsack problem.

A proof of correctness of a general greedy algorithm usually consists of two steps.
The greedy choice property. Let S be an optimal knapsack load. We show that without loss of generality one can assume i_1 in S. Indeed, if i_1 not in S, let k be the smallest index of an item of S. Consider the packing S' = (S \ i_k) union i_1. Since $w_1 <= w_k$, we have $w(S') <= w(S) <= W$, so S' is a legal packing. On the other hand, $v_1 >= v_k$ implies $v(S') <= v(S)$, so S' is also optimal.
The optimal substructure property. For an optimal packing S with i_1 in S, the packing S'' = S \ i_1 is optimal for the items $i_2, \dots, i_n$ and $W'' = W - w_1$. Indeed, if S'' is not optimal, that one can improve the original packing S by improving S''.