1- [20 pts] Provide a short answer to the following questions.

(a) Draw the graph, given by the following formal definition:

*G* = (*V, A*)

*V* = *{a, b, c, d, e}*

*A* = *{*(*a, c*)*,* (*b, c*)*,* (*c, a*)*,* (*c, d*)*,* (*d, b*)*,* (*d, e*)*,* (*e, a*)*,* (*b, a*)*}*

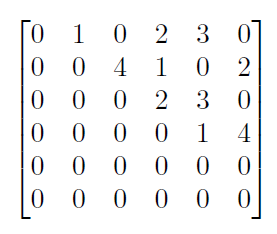
(b) What is one quality that all trees exhibit, that graphs, in general, do not?

(c) What is hashing good for?

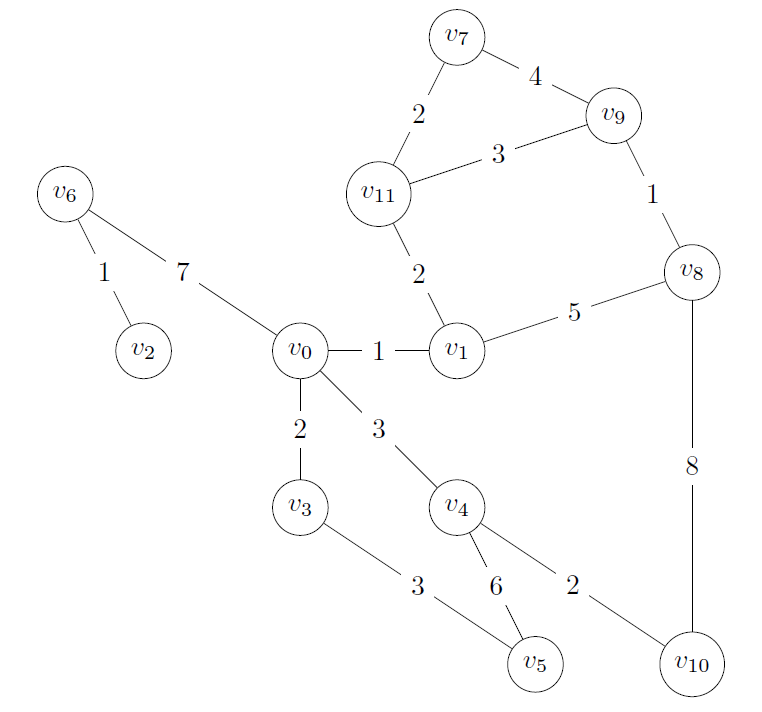
(d) What is the result (table) of the following join operation:

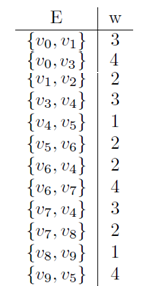
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| SELECT *Col1, Col3, Col5*  FROM *T1, T2*  WHERE *Col2 = Col4* | |  |  | | --- | --- | | Col1 | Col2 | | A | 10 | | B | 12 | | C | 8 |   **T1:** | |  |  |  | | --- | --- | --- | | Col3 | Col4 | Col5 | | D | 12 | XY | | E | 8 | YZ | | F | 9 | WX |   **T2:** |

2- (10 pts) (a) Given the following adjacency matrix, draw the weighted, undirected graph with

*V = {v0, v1, v2, v3, v4, v5}.*

(b) What is a topological sort for this graph?

3- (20 pts) Use Kruskal’s algorithm to construct a minimum spanning forest for the following graph. Yes, you may the draw graphs. Show all steps and the final cost.

****Question 4 (20 Points)** Use Prim’s Algorithm to calculate the minimum spanning tree of the following graph *G* = (*V, E, w*). The source vertex is *v0*. Show all steps and the final cost. *V* = *{v0, v1, v2, v3, v4, v5 , v6, v7, v8 , v9 }*

*{V0, V1}: 3*

*{V1, V2}: 2*

*{V0, V3}: 4*

*{V3, V4}: 3*

*{V4, V5}: 1*

*{V4, V6}: 2*

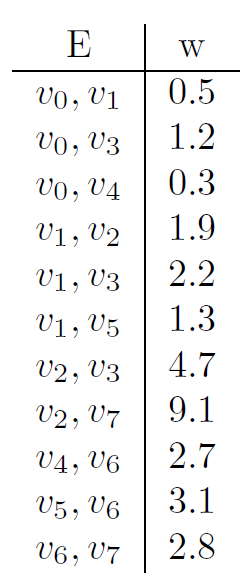
*{V4, V7}: 3*

*{V7, V8}: 2*

*{V8, V9}: 1*

*21*

**Question 5 (20 Points)** Given the graph *G = (V, E, w)*, below, find the shortest paths between *v0* and all other vertices.

*V* = *{v0, v1, v2, v3, v4, v5 , v6, v7 }*

6- (10 pts) Build a *Hash Table* of size 8 by inserting the following values in order. Use the algorithm provided to calculate hash values, and open addressing (linear probing) to resolve all collisions. Show all intermediate tables, include both the value and its index in your illustrations.

12, 456, 137, 10907, 1144, 953, 4, 3713

int hash (int input) { int h = input  10;

h += (input / 10)  10;

return h  8;

}

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| SELECT *Col1, Col3, Col5*  FROM *T1, T2*  WHERE *Col2 = Col4* | |  |  | | --- | --- | | Col1 | Col2 | | A | 10 | | B | 12 | | C | 8 |   **T1:** | |  |  |  | | --- | --- | --- | | Col3 | Col4 | Col5 | | D | 12 | XY | | E | 8 | YZ | | F | 9 | WX |   **T2:** |