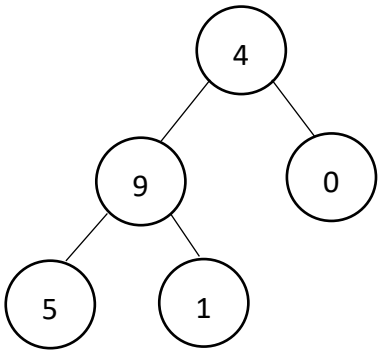
Algorithm Spring 2021 Final Exam Solution

1. (10%) Considered a binary tree which contains digits from $0-9$ only, each root-to-leaf path would represent a number. If the leaf is the left-child or has no sibling, the number should be positive; otherwise, it should be negative. For instance, for a binary tree below, a root-to-leaf path $4 \to 9 \to 5$ which represents the number 495, and a root-to-leaf path $4 \to 9 \to 1$ which represents the number -491. Please answer the following questions.
   Input: $[4, 9, 0, 5, 1]$ (Suppose that the first index is 0)



(a) (5%) Given pseudo code for the problem that finding the total sum of all root-to-leaf numbers. Please fill in blanks. (Assume that input array is T.)

```
1.  dfs(Tree, i, num) {
2.      // Calculate value to Tree[i]
3.      val = ____(1)____
4.      // If there is no child, add the value of root-to-leaf to global variable "sum"
5.      if(Tree[i*2+1] == null && Tree[i*2+2] == null){
6.          // If leaf is right child, then multiply the value of root-to-leaf with (-1)
7.          if(____(2)____)
8.              val = val*(-1)
9.          sum = sum + val
10.         return
11.     }
12.     // search for left child
13.     if(Tree[i*2+1] != null)
14.         ____(3)____
15.     // search for right child
16.     if(Tree[i*2+2] != null)
17.         ____(4)____
18. }
```

```
19.
20.  sum = 0 // global variable
21.  sumNumbers(Tree, root) {
22.      if(Tree[root] == null)
23.          return 0
24.      dfs(Tree, root, 0)
25.      return sum
26.  }
27.
28. main() {
29.     print(sumNumbers(T, 0))
30. }
```

(b) (5%) What output will be printed if T is $[2, 7, 8, 0, 1, 3]$?

Ans:

(a) (1) num*10 + Tree[i]
(2) i%2 == 0 && i != 0
(3) dfs(Tree, i*2+1, val)
(4) dfs(Tree, i*2+2, val)

(b) 270-271+283 = 282

2. (10%)
(a) (5%) Why doesn't Dijkstra's algorithm work for the graph that has negative path?
(b) (5%) Given Dijkstra's algorithm below.

INITIALIZE-SINGLE-SOURCE(G, s)

```
1 for each vertex v ∈ G.V do
2     v.d = ∞
3     v.π = NIL
4 s.d = 0
```

RELAX(u, v, w)

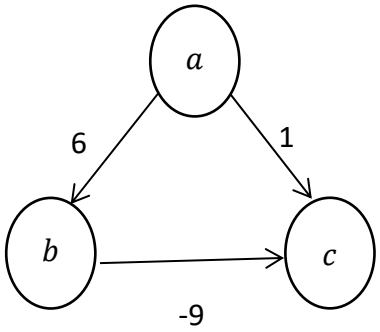
```
1 if v.d > u.d + w(u, v) then
2     v.d = u.d + w(u, v)
3     v.π = u
```

```
DIJKSTRA(G, w, r)
1 INITIALIZE-SINGLE-SOURCE(G, s)
2 S = ∅
3 Q = G.V
4 while Q ≠ ∅ do
5     u =EXTRACT-MIN(Q)
6     S = S ∪ {u}
7     for each vertex v ∈ G.Adj[u] do
8         RELAX(u, v, w)
```

Analyze and give time complexity for Dijkstra's algorithm while using binary heap. Suppose the input graph G is dense, which means that the graph can be assumed as complete graph. (You need to show your reason.)

Ans:

(a) Note that in Dijkstra's algorithm, **once a vertex is marked as "closed", the algorithm found the shortest path to it**, and will never have to develop this node again - it assumes the path developed to this path is the shortest. However, it might not be true with negative weights. For example:



It will fail to find the shortest path $a \to b \to c$ with total weight -3. (Assume started from a)

(b) $\theta(ElgV)$

Let V be number of vertices, E be number of total edges. Suppose Q is a binary heap. Next, we analyze the time complexity:

1. Initialize Q and first for loop: $O(VlgV)$ and $O(V)$.
2. DECREASE-KEY of r: $O(lgV)$.
3. EXTRACT-MIN in while loop: $O\big((1+2+\cdots+(V-1)) * lgV\big)$

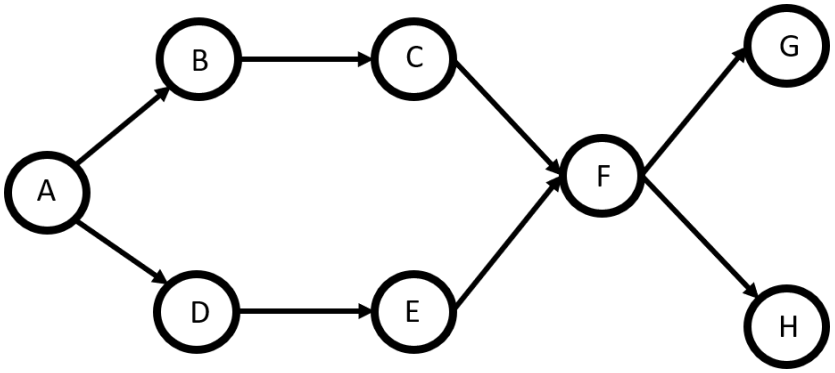
$= O(\Big(V * \frac{V-1}{2}\Big) * lgV\ ) \approx O(V^2 lgV)\ \approx O(ElgV)$.

(Since the graph is dense.)

Therefore, the total time complexity would be

$$O(VlgV) + O(V) + O(lgV) + O(ElgV) \;=\; O(ElgV)$$

3. (10%)



Consider the directed acyclic graph G above. How many topological orderings does it have?

Ans:

$$A \cdot \{BC \cdot DE\} \cdot F \cdot \{G \cdot H\} = (2 + 2 \times 2) \times 2 = 12$$

A BCDE F {GH}
A DEBC F {GH}
A {BD}{CE} F {GH}

4. (10%)

(a) (5%) Given 10 difference characters and their frequency, please build a Huffman tree.

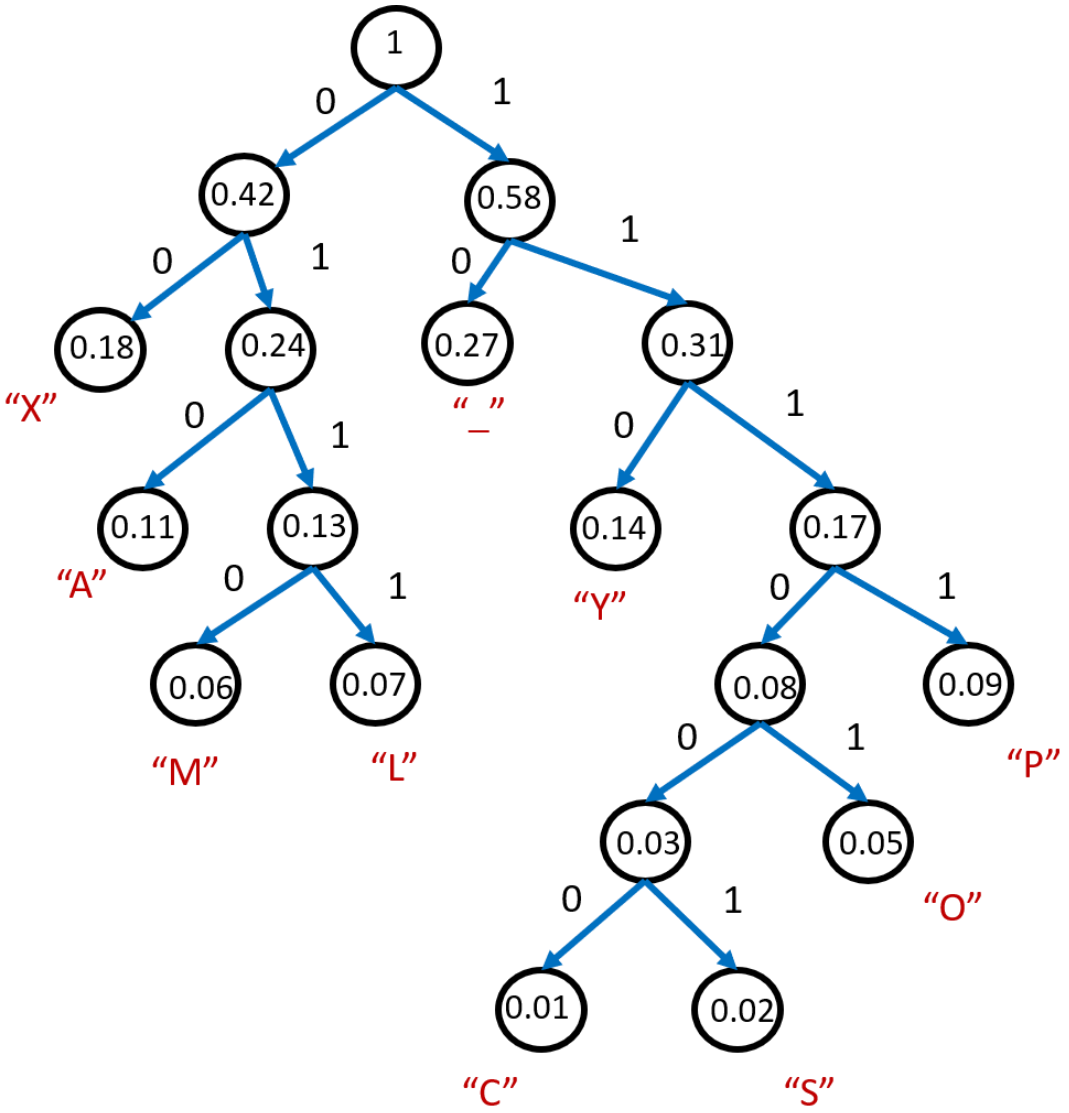
Character	"C"	"O"	"S"	"M"	"_"	"P"	"A"	"X"	"Y"	"L"
Frequency	0.01	0.05	0.02	0.06	0.27	0.09	0.11	0.18	0.14	0.07

(Note 1: All edges along the path to a character contain a code digit. If they are on the left side of the tree, they will be a 0 (zero). If on the right, they well be a 1 (one).)
(Note 2: the frequency of left child < frequency of right child at the same height)

(b) (5%) Following the previous question (a), given a string:
0100111011110111101011100111100 1
Please decode the string by your Huffman tree.

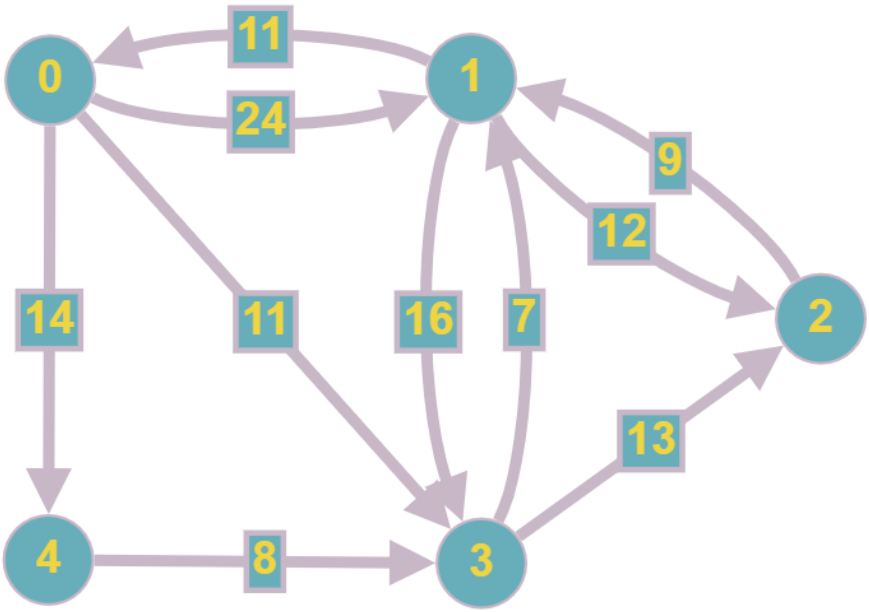
Ans:

(a)



2.(b) ALL_PASS

5. (10%)



Find all pairs shortest path using Floyd-Warshall.
(Please write down process)

Ans:

initial

	0	1	2	3	4
0	0	24	X	11	14
1	11	0	12	16	X
2	X	9	0	X	X
3	X	7	13	0	X
4	X	X	X	8	0

0

	0	1	2	3	4
0	0	24	X	11	14
1	11	0	12	16	25
2	X	9	0	X	X
3	X	7	13	0	X
4	X	X	X	8	0

1

	0	1	2	3	4
0	0	24	36	11	14
1	11	0	12	16	25
2	20	9	0	25	34
3	18	7	13	0	32
4	X	X	X	8	0

2

	0	1	2	3	4
0	0	24	36	11	14
1	11	0	12	16	25
2	20	9	0	25	34
3	18	7	13	0	32
4	X	X	X	8	0

3

	0	1	2	3	4
0	0	18	24	11	14
1	11	0	12	16	25
2	20	9	0	25	34
3	18	7	13	0	32
4	26	15	21	8	0

4

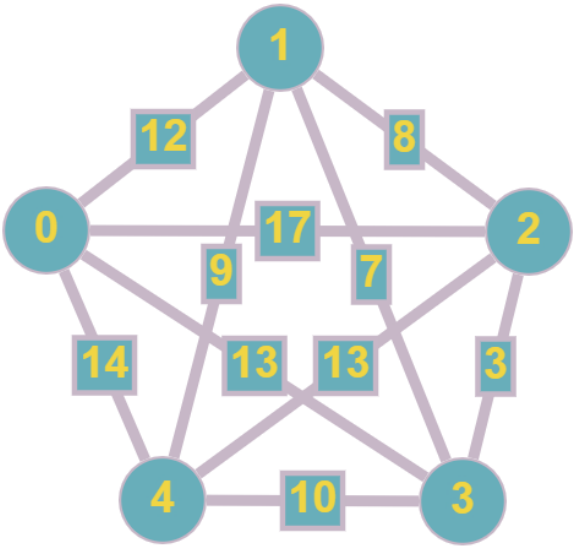
	0	1	2	3	4
0	0	18	24	11	14
1	11	0	12	16	25
2	20	9	0	25	34
3	18	7	13	0	32
4	26	15	21	8	0

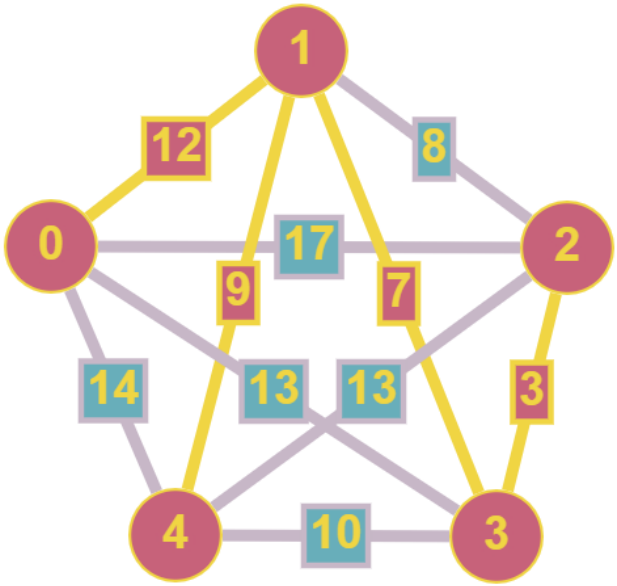
6. (10%) There are five cities in a network. The cost of building a road directly between i and j is the entry $a_{i,j}$ in the matrix below. Determine the least cost of making all the cities reachable from each other.

	0	1	2	3	4
0	0	12	17	13	14
1	12	0	8	7	9
2	17	8	0	3	13
3	13	7	3	0	10
4	14	9	13	10	0

Ans:

Weight of minimum spanning tree is 31.





7. (10%) Given a directed graph $G = (V, E)$, please design an algorithm to determine whether it contains a (directed) cycle.

Ans:

DFS(G), if there exist a back edge then return “cycle”; else return “no cycle”.

8. (10%) Given a directed graph $G = (V, E)$, please design an algorithm to determine whether it contains a (directed) cycle of odd length.

Ans:

First find out the strongly connected components, and perform the following steps for each component

1. Perform DFS on any point r, and mark r as even
2. Every time you visit a point, let it mark it in the opposite direction to the previous point. If this point has been marked and is opposite to the direction we want to mark, it will have a cycle of odd length.

9. (10%) Given a set of tasks, each task requests one unit of time to finish work. Moreover, a task has a deadline and reward. If the task is finished before the deadline, we get the reward. We cannot schedule multiple tasks at same time. (Hint: the deadline is at least one.)

   (a) (5%) Please design an algorithm to maximize total reward.
   (b) (5%) Consider any nonempty subproblem S_k, and let t_m be a task in S_k that is according to your choice. Please show that t_m is included in some maximum total reward subset of mutually compatible tasks of S_k.

Ans:

1) Sort all tasks in decreasing order of reward.
   Iterate on tasks in decreasing order of reward. For each task, do the following : For each task find an empty time slot from deadline to 0. If found empty slot put the task in the slot and mark this slot filled.

2) Suppose that we had some other solution S_k that do not include t_m, we could substitute the higher reward task t_m for a lower reward task before the deadline of t_m meaning our original solution could not have been optimal.

Examples:

Input: Four Jobs with following deadlines and reward

JobID	Deadline	reward
a	4	20
b	1	10
c	1	40
d	1	30

Output: Following is maximum reward sequence of jobs

c, a

Input: Five Jobs with following deadlines and reward

JobID	Deadline	reward
a	2	100
b	1	19
c	2	27
d	1	25
e	3	15

Output: Following is maximum reward sequence of jobs

c, a, e

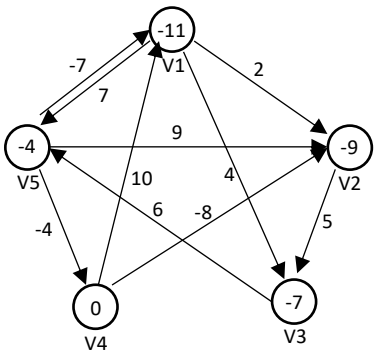
10. (10%) Consider the following linear-programming system of difference constraints:

$x_3 - x_1 \leq 4$
$x_2 - x_1 \leq 2$
$x_1 - x_4 \leq 10$
$x_5 - x_1 = 7$
$x_3 - x_2 \leq 5$
$x_2 - x_4 \leq -8$
$x_4 - x_5 \leq -4$
$x_5 - x_3 \leq 6$
$x_2 - x_5 \leq 9$

(a) (5%) Draw the constraint graph for these constraints.
(b) (5%) Find a feasible solution or explain why no feasible solution exists.

Ans:

1)



2)
Because there is a negative cycle, we can't find a feasible solution.