Grade: 8/9

Exercise 4 (for grade) ~ Wednesday, September 28, 2022 ~ CPSC 535 Fall 2022

(+0.5 points): bonus for 2-people group

Write one submission for your entire group, and write all group members' names on that submission. Turn in your submission before the end of class. The X symbol marks where you should write answers.

> Recall that our recommended problem-solving process is:
> 1. **Understand** the problem definition. What is the input? What is the output?
> 2. **Baseline** algorithm for comparison
> 3. **Goal** setting: improve on the baseline how?
> 4. **Design** a more sophisticated algorithm
> 5. **Inspiration** (if necessary) from patterns, bottleneck in the baseline algorithm, other algorithms
> 6. **Analyze** your solution; goal met? Trade-offs?

Follow this process for each of the following computational problems. For each problem, your submission should include:

a. State are the input variables and what are the output variables
b. Pseudocode for your baseline algorithm, that needs to include the data type and an explanation for any variable other than input and output variables
a. The Θ-notation time complexity of your baseline algorithm, with justification.

and if you manage to create an improved algorithm:

c. Answer the question: how is your improved algorithm different from your baseline; what did you change to make it faster?
d. Pseudocode for your improved algorithm, that needs to include the data type and an explanation for any variable other than input and output variables
a. The Θ-notation time complexity of your improved algorithm, with justification.

Today's problems are:

1.

Given a binary search tree with n nodes ($n > 3$) and a query value q, design an algorithm that returns NONE if the query value is not in the tree, or the depth of the node that holds the query value. The depth of the root node is 0.

Baseline: time complexity O(n), improved to O(height) time complexity

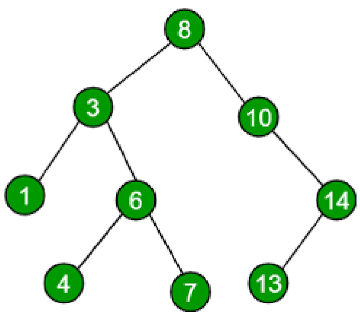


Fig. 1 Image taken from https://www.geeksforgeeks.org/check-if-given-sorted-sub-sequence-exists-in-binary-search-tree/

Example 1:	Input: BST in Figure 1, q = 14	Output: 2
Example 2:	Input: BST in Figure 1, q = 7	Output: 3
Example 3:	Input: BST in Figure 1, q = 8	Output: 0
Example 3:	Input: BST in Figure 1, q = 9	Output: NONE

2.

a) Run the Floyd-Warshall algorithm on the weighted, directed graph shown below. Show the matrix $D^{(k)}$ that results for each iteration of the outer loop. Show your work at every step. I will deduct points if you do not show your work and you skip steps.

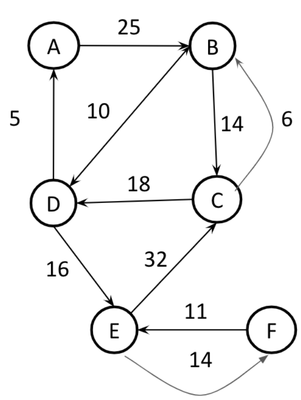


Fig. 2 Example of weighted, directed graph

b) State the dimensionality of the Floyd-Warshall algorithm as defined for a dynamic programming algorithm.

3.

Compute the transitive closure of the graph above. Show your work at every step. I will deduct points if you do not show your work and you skip steps.

Names

Write the names of all group members below.

☒Kevin Jasani, Rosa Cho,

Exercise 1: Solve and provide answer

☒For finding element in binary search tree, we need to do traversal from root to end node.

```
Start :  // create a function for searching elemet
         // initialisation of variable for counting
         int countBase=0;
         int treeHeight(Node root)
         {
                  if(root==null)
                  {
                           return ;
                  }
                  Else
                  {
                           int  lheight= treeHeight(root.left);
                           int  rheight= treeHeight(root.right);
                           return Math.max(lHeight,rheight);
                  }
         }

         int height= treeHeight(root)
         Public Node SearchBinaryTree (Node root, int key, int countBase)
         {
                  //if root is null or we will get key on root node
                  if(root==null || root.key==key)
                  {
                           return root;
                           countBase=height-countBase;
                           System.out.prinrtln(" Element is found on depth :"+countBase);
                  }
                  // if key is less then root's Key;
                  if( root.key>key)
                  {
                           countBase++;
                           return search(root.left, key,countBase);
                  }else if( root.key<key)
```


(-1 point): incorrect algorithm; depth of a node is not equal to the height of the tree minus the distance to the root, but the distance to the root

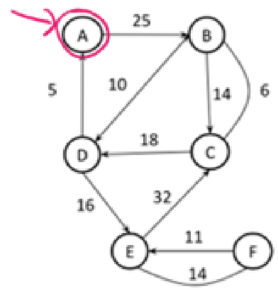
```
        {
            countBase++;
            return  search (root.right, key,countBase);
        }
    }
```

Exercise 2: Solve and provide answer

X

2 a)

1. Inputs:
   Number of vertices: 6
   Output: Find all pair Shortest path

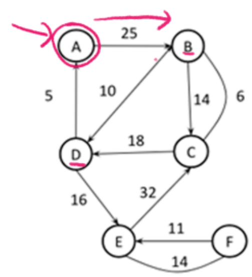


2. D^0

	A	B	C	D	E	F
A	0	25	∞	∞	∞	∞
B	∞	0	14	∞	∞	∞
C	∞	∞	0	18	32	∞
D	5	∞	∞	0	16	∞
E	∞	∞	32	∞	0	14
F	∞	∞	∞	∞	11	0

D_1

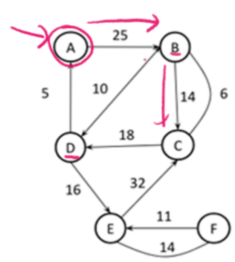
	A	B	C	D	E	F
A	0	25	∞	15	∞	∞
B	20	0	14	10	∞	∞
C	∞	∞	0	18	32	∞
D	5	∞	∞	0	16	∞
E	∞	∞	32	∞	0	14
F	∞	∞	∞	∞	11	0



D_2

+Win

	A	B	C	D	E	F
A	0	25	∞	15	∞	∞
B	20	0	14	10	∞	∞
C	∞	10	0	18	34	∞
D	5	∞	30	0	16	∞
E	∞	∞	32	∞	0	14
F	∞	∞	∞	∞	11	0



$[C,E] = D^2(C,E) + (C,E)$

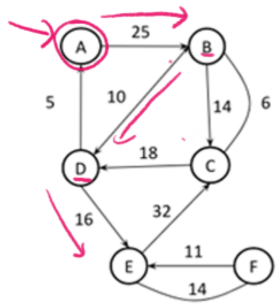
$[18+16+32 = (?) \pm 32 = 34 < \infty$

$[D,C] = [D,C] + [C,D]$

$[16+32^{48}] - 18 = 30$

D_3

	A	B	C	D	E	F
A	0	25	∞	-5	∞	∞
B	-20	0	14	10	∞	∞
C	∞	10	0	18	34	∞
D	5	∞	30	0	16	∞
E	∞	9	32	66	0	14
F	∞	∞	∞	∞	11	0

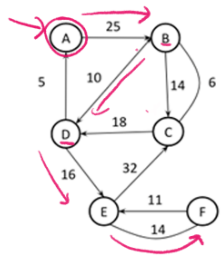


[D,E] = [D,E] + [E,D]

[16] + [$\overset{50}{32+18}$] = 66

D_4

	A	B	C	D	E	F
A	0	25	∞	-5	∞	0
B	-20	0	14	10	∞	121
C	∞	10	0	18	34	89
D	5	∞	30	0	16	91
E	∞	9	32	66	0	14
F	∞	∞	∞	∞	11	0



[F,D] = [D,F] + [F,D]

(16+14) + [11+32+18]

30 + 61 = 91

[F, C] = [C,F] + [F,C]

(32+14) + (11+32)

46 + 43 = 89

[F,B] = [B,F] + [F,B]

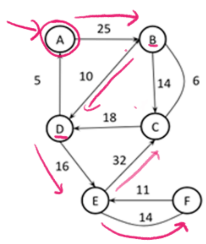
(11+32+14)

51 + [14+1]

62 + 16+14

D

	A	B	C	D	E	F
A	0	25	∞	-5	16	0
B	-20	0	14	10	86	121
C	∞	10	0	18	34	89
D	5	∞	30	0	16	91
E	∞	9	32	66	0	14
F	∞	∞	∞	∞	11	0



[B,B] = [B,C] + [C,B]

(14+18+1) + (2+6)

48 + 38

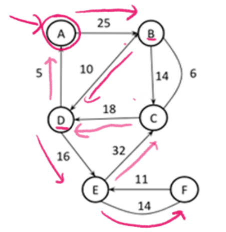
[E,A] = [A,E] + [E,A]

(25+14+16) + (32+18+5)

51 + 55

D

	A	B	C	D	E	F
A	0	25	62	-5	166	0
B	-20	0	14	10	86	121
C	∞	10	0	18	34	89
D	5	62	30	0	16	91
E	72	87	32	66	0	14
F	55	32	10	0	11	0



[B,D) = (B,D) + [A,B]
(14+18) + [5+25]
32 + 30 = 62

(A,C] = [A,C] + [C,A]
(25+14) + [18+5]
39 + 23 = 62

(-0.25 points): last step is missing; there are 6 nodes not 5

2. b) The dimensionality of the algorithm as stated through dynamic programming is 6x6.

(-0.25 points): incorrect dimensionality

Exercise 3: Solve and provide answer

DATE :

(A,B) (B,C) (C,B) (D,A) (B,D) (D,E) (C,D),
(E,C), (E,F), (F,E)

$$
\begin{array}{c|cccccc}
 & A & B & C & D & E & F \\
\hline
T_0 : A & 1 & 1 & 0 & 0 & 0 & 0 \\
B & 0 & 1 & 1 & 1 & 0 & 0 \\
C & 0 & 1 & 1 & 1 & 0 & 0 \\
D & 1 & 0 & 0 & 1 & 1 & 0 \\
E & 0 & 0 & 1 & 0 & 1 & 1 \\
F & 0 & 0 & 0 & 0 & 1 & 1
\end{array}
$$

$C_1 \times R_1$

$= C_1 = \{A, D\}$
$R_1 = \{A, B\}$
$= (A,A) (A,B) (D,A) (D,B)$

$$
\begin{array}{c|cccccc}
 & A & B & C & D & E & F \\
\hline
T_1 = A & 1 & 1 & 0 & 0 & 0 & 0 \\
B & 0 & 1 & 1 & 1 & 0 & 0 \\
C & 0 & 1 & 1 & 1 & 0 & 0 \\
D & 1 & 1 & 0 & 1 & 1 & 0 \\
E & 0 & 0 & 1 & 0 & 1 & 1 \\
F & 0 & 0 & 0 & 0 & 1 & 1
\end{array}
$$

$C_2 \times R_2$

$= C_2 = \{A, B, C, D\}$
$R_2 = \{B, C, D\}$

$= (A,B) (A,C) (A,D)$
$(B,B) (B,C) (B,D)$
$(C,B) (C,C), (C,D)$
$(D,B), (D,C), (D,D)$

$$
\begin{array}{c|cccccc}
 & A & B & C & D & E & F \\
\hline
T_2 = A & 1 & 1 & 1 & 1 & 0 & 0 \\
B & 0 & 1 & 1 & 1 & 0 & 0 \\
C & 0 & 1 & 1 & 1 & 0 & 0 \\
D & 1 & 1 & 1 & 1 & 1 & 0 \\
E & 0 & 0 & 1 & 0 & 1 & 1 \\
F & 0 & 0 & 0 & 0 & 1 & 1
\end{array}
$$

$C_3 \times R_3$

$= C_3 = \{A, B, C, D, E)$
$= R_3 = \{B, C, D\}$

$= (A,B) (A,C) (A,D)$
$(B,B) (B,C) (B,D)$
$(C,B), (C,C), (C,D)$
$(D,B) (D,C), (D,D)$
$(E,B) (E,C) (E,D)$

T_3:

$$\begin{array}{c|cccccc} & A & B & C & D & E & F \\ \hline A & 1 & 1 & 1 & 1 & 0 & 0 \\ B & 0 & 1 & 1 & 1 & 0 & 0 \\ C & 0 & 1 & 1 & 1 & 0 & 0 \\ D & 1 & 1 & 1 & 1 & 1 & 0 \\ E & 0 & 1 & 1 & 1 & 1 & 1 \\ F & 0 & 0 & 0 & 0 & 1 & 1 \end{array}$$

$C_4 \times R_4$
$C_4 = (A, B, C, D, E)$
$R_4 = \{A, B, C, D, E\}$

= (A,A) (A,B) (A,C) (A,D) (A,E)
(B,A) (B,B) (B,C) (B,D) (B,E)
(C,A), (C,B), (C,C) (C,D), (C,E)
(D,A), (D,B), (D,C), (D,D), (D,E)
(E,A), (E,B), (E,C), (E,D) (E,E)

T_4:

$$\begin{array}{c|cccccc} & A & B & C & D & E & F \\ \hline A & 1 & 1 & 1 & 1 & 1 & 0 \\ B & 1 & 1 & 1 & 1 & 1 & 0 \\ C & 1 & 1 & 1 & 1 & 1 & 0 \\ D & 1 & 1 & 1 & 1 & 1 & 0 \\ E & 1 & 1 & 1 & 1 & 1 & 1 \\ F & 0 & 0 & 0 & 0 & 1 & 1 \end{array}$$

$C_5 \times R_5$
$C_5 = \{A, B, C, D, E$
$R_5 = \{A, B, C, D, E$

(A,A) (A B) (A C) (A D) (A E)
(B,A) (B,B) (B,C), (B,D), (B,E)
(C,A) (C B) (C,C) (C,D) (C,E)
(D,A) (D,B) (D,C) (D,D) (D,E)
(E A) (E B) (E C) (E D) (E E)
(F A) (F,B) (F,C) (F,D) (F,E)

PAGE NO.
DATE :

T_5:

$$\begin{array}{c|cccccc} & A & B & C & D & E & F \\ \hline A & 1 & 1 & 1 & 1 & 1 & 1 \\ B & 1 & 1 & 1 & 1 & 1 & 1 \\ C & 1 & 1 & 1 & 1 & 1 & 1 \\ D & 1 & 1 & 1 & 1 & 1 & 1 \\ E & 1 & 1 & 1 & 1 & 1 & 1 \\ F & 1 & 1 & 1 & 1 & 1 & 1 \end{array}$$

$C_6 \times R_6$

$C_6 = \{A, B, C, D, E, F\}$

$R_6 = \{A, B, C, D, E, F\}$

Ear

Final Matrix

T_6

$$\begin{array}{c|cccccc} & A & B & C & D & E & F \\ \hline A & 1 & 1 & 1 & 1 & 1 & 1 \\ B & 1 & 1 & 1 & 1 & 1 & 1 \\ C & 1 & 1 & 1 & 1 & 1 & 1 \\ D & 1 & 1 & 1 & 1 & 1 & 1 \\ E & 1 & 1 & 1 & 1 & 1 & 1 \\ F & 1 & 1 & 1 & 1 & 1 & 1 \end{array}$$