Design and Analysis of Algorithms CS 575, Spring 2023 Theory Assignment 3.1

- 1. The edit distance between two strings S1 and S2 is the minimum number of operations to convert one string to the other string. We assume that three types of operations can be used: Insert (a character), Delete (a character), and Replace (a character by another character). For example, the edit distance between dof and dog is 1 (one Replace), between cat and act is 2 (one Delete and one Insert or two Replace), between cat and dog is 3 (3 Replace). Design a dynamic programming algorithm to compute the edit distance between two strings by following the steps below:
 - a. [10 points] Write down the principle of optimality for the minimum edit distance problem, and prove that the problem satisfies the principle of optimality.

- The principle of optimality states that an optimal solution to a problem contains within it
 optimal solutions to subproblems.
- In the case of minimum edit distance, this means that the minimum edit distance between two strings S1 and S2 can be computed by finding the minimum edit distance between prefixes of S1 and S2, and using those minimum edit distances to compute the minimum edit distance between S1 and S2.
- For example, suppose we want to find the minimum edit distance between the strings "cat" and "bat". We can break the problem down into smaller subproblems by considering the prefixes of the two strings:
 - > The minimum edit distance between the prefixes "" and "" is 0.
 - The minimum edit distance between the prefixes "c" and "" is 1 (one deletion).
 - ➤ The minimum edit distance between the prefixes "c" and "b" is 1 (one substitution).
 - The minimum edit distance between the prefixes "ca" and "b" is 2 (one substitution and one deletion).
 - The minimum edit distance between the prefixes "cat" and "b" is 3 (one substitution and two deletions).
- Using the minimum edit distances between prefixes, we can compute the minimum edit distance between "cat" and "bat" as 1 (one substitution). This shows that the problem satisfies the principle of optimality, as the optimal solution to the problem contains within it optimal solutions to subproblems.

b. [10 points] Show the recurrence equation for computing the edit distance. (Hint: Let d[i, j] be the edit distance between the substring of the first i characters of S1 and the substring of the first j characters of S2. Then consider the prefixes of the two strings in a way similar to the analysis for the LCS problem.)

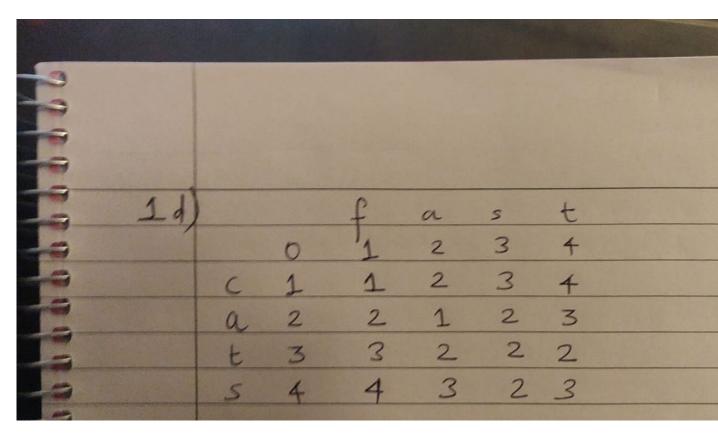
THE RESERVE THE PERSON NAMED IN COLUMN TWO IS NOT THE OWNER.	
7 ()	Recovence Equation:
	9
	eD(i,j)=
	$eD(i-1,j-1)$ $y S_1[i] = S_2[j]$
	min $\int eD(i,j-1)+1$
	$\begin{cases} eD(i-1,j)+1 & \text{of then } Cost = 0 \\ eD(i-1,i-1)+cost & \text{else } Cost = 1 \end{cases}$

c. [10 points] Provide pseudocode for Edit-Distance(S1, S2).

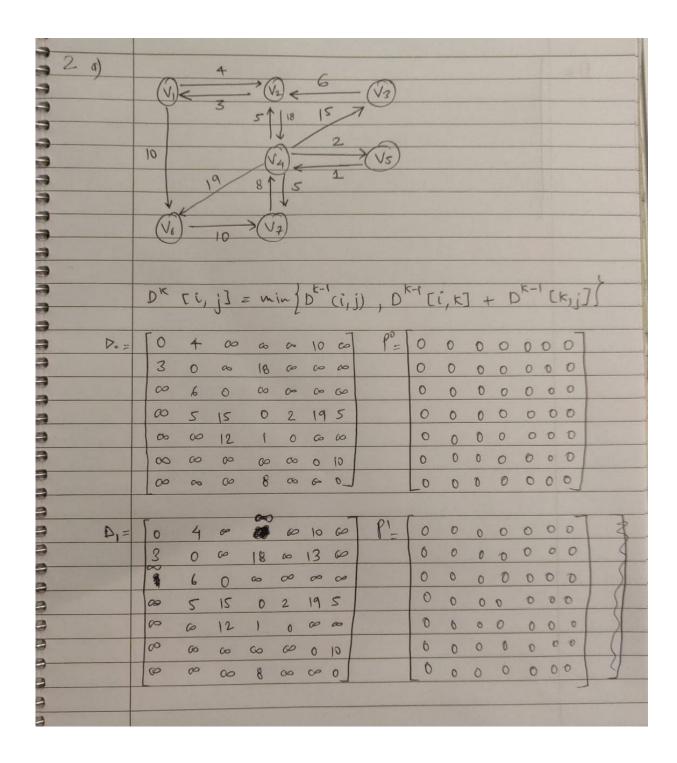
Solution:	
150	int edit Distance (S1, S2)
1+0)	int earthistance (SI, SZ)
	2 = length (SI)
	y = length (52)
	y = length(S2) eD [x+1] [y+1]
	Cor i = 0 to x do:
	for i = 0 to x do: eD [i][6] ← i
	for j = 0 by do: eD [o][j] = j
	ev lost je
	THE STORY AS SECURED WANTE OF A SHARE WELL AS A SHARE WAS A SHARE
	for i = 1 to x do:
	for j = 1 to y do:
	$y s_1 [i-1] == s_2 [j-1] do:$ $eD [i] [j] \leftarrow eD [i-1] [j-1]$
	$y_{S1}[i-1] = = S2[j-1] do:$ $eD[i][j] \leftarrow eD[i-1][j-1]$
	60503113
	else do:
4047-8E	ed [ilj] = 1+ min(ed[i][j-1], ed [1-1][j]
	ed [i][j] < 1 + min(ed[i][j-1], ed[i-1][j-1], ed[i-1][j-1]
	•
	retorn eD [x][y]

d. [10 points] Use Edit-Distance() to create the table d (d[i, j] is defined above) for S1 = cats and S2 = fast. The entry at d[4, 4] should show the correct edit distance between the two words.

Solution:



- 2. [35 points] Use Floyd's algorithm to find all pairs shortest paths in the following graph.
 - a. [15 points] construct the matrix D, which contains the lengths of the shortest paths, and the matrix P, which contains the highest indices of the intermediate vertices on the shortest paths. Show the actions step by step. You need to show D⁰ to D⁷ and P⁰ to P⁷ (i.e. matrix P updated along with D step by step). You can use your computer program to output them or do it manually.



9934	2-14		14	200	1 24		6331	WEEK.	113		4	337	_	_/_	/	
	-							2 0 1	100						-	-
02	0	4	0	22	60	10	00	P=	0	0	0	2	0	0	0	-
	3	0	00	18	00	15	3 00		0	0	0	0	0	0	0	
	9	6	0	24	00	10	1 00		2	0	0	2	0	2	0	6-18
	8	5	15	0	2	1	8 5		2	0	0	0	0	2	0	5
	00	00	12	1	0	0	00		0	0	0	0	0	0	0	0
	0-	00	80	0	0-	0) 10	1 335	0	0	0	0	0	0	0	2
1500	00	00	00	8	00	0	0 0		0	0	0	0	0	0	0	2
T. C. L. of	-															
Ds	0	4	00	22	60	10	0	7 7 =	0	0	0	2	0	0	0	0
	3	0	00	18	00	13	Go		0	0	0	0	0	0	O	0
	9	6	0	24	60	10	00	Bar	2	0	0	2	0	2	0	0
	8	5	15	0	2	(8	2		2	0	0	0	0	2	0	0
D. T.B.	21	18	12	1	0	31	00		3	3	0	0	0	3	0	0 3
	00	00	00	00	00	0	10		0	0	0	0	0	0	0	0 3
	00	60	O	8	OP.	00	0	W- 142	0	0	0	0	0	0	0	0.0
																0.5
							- 1		PK							
04	50	4	37	22	24	10	27	P4=	0	0	4	2	4	0	4	0
By	0	9	37 33	22	24	10	23	P4=	0	0	4	2			4	0
Øı		0	33			13	Jane 1	P4=	0	0	1	0	4 4 4	0	4	0.00
By	3	0	11000	(8	20	13	23	P =	0 2	0	4	0	4	0	_	000
Øy	3	0 6 S	33 0	18	26	13	23	P =	0	0	4	2 0	4 9 0	0	4	
Øy	3 9 8	0 6 5	33	18	26 2 0	13 19 19	23 29 5	P =	0 2 2	0 0	4 0	0 0	4 9 0	0 2 2	4 0	
By	9 9	0 6 5 6	33 0 15 12	18 24 0	20 26 2 0	13	23 29 5	P =	0 2 2 9	0 0 0 9	4 0 0	2 0	4 9 0	0 2 2 4	4 0 4	
Ø _Y	3 9 8 9	0 6 5	33 0 15	18 24 0 1	26 2 0	13 19 13 19 0	23 29 5 6	P =	0 2 2 9 0	0 0 0 4	4 0 0 0	0 0 0	4 4 0 0 0	0 2 2 4 0	4 0 4 0	
	2 1 8 9 0 16	0 6 5 6 6 6 13	33 0 15 12 0 23	18 24 0 1 0 8	20 26 2 0 0	13 19 19 0 26	23 29 5 6 10		0 2 2 9 9	0 0 0 9	4 0 0 0 0 4	0 0 0 0	4 4 0 0 0	0 2 2 4 0	4 0 4 0	
Ps	9 60 16	0 6 5 6 6 13	33 0 15 12 0 23	18 24 0 1 0 8	20 26 2 0 0 10	13 19 13 19 0 26	23 29 5 6 10 0	P5=	0 2 2 9 0 9	0 0 0 9 0 9	4 0 0 0 0 4	0 0 0 0	4 4 0 0 0 0 4 4	0 2 2 4 0 4	4 0 4 0 0	
	9 8 9 66 16	0 6 5 6 60 13	33 0 15 12 0 23 36 32	18 24 0 1 0 8	20 26 2 0 0 10 24 20	13 19 13 19 0 26	23 29 5 6 10 0		0 2 2 9 0 9	0 0 0 9 9 0 0	400004	0 0 0 0 0 2 0	440004 44	0 2 2 4 0 4	4 0 4 0 0 4 4	
	9 8 9 60 16	0 6 8 13 4 0 6	33 0 15 12 0 23 36 32 0	18 24 0 1 0 8 22 18 24	20 26 2 0 0 10 24 20 26	13 19 19 0 26 10 13	23 29 5 6 10 0] 24 23 29		0 2 2 9 0 9	0 0 0 9 9 0 0 0 0	400004550	0 0 0 0 0 2 0 2	470004 777	0 2 2 4 0 4 0 0 0 2	4 0 4 0 0	
	9 8 9 6 16	0 6 8 6 80 13 4 0 6 5	33 0 15 12 23 23 36 32 0	18 24 0 1 6 8 22 18 24 0	20 26 2 0 0 10 24 20 26 2	13 19 0 26 10 13 19	23 29 5 6 10 0] 27 23 29 5		0 2 2 4 0 4 0 0 2 2	0 0 0 9 9 0 0 0 0 0	4 0 0 0 0 4 5 5 0 5	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	440004 4440	0 2 2 4 0 4 0 0 2 2	4 0 4 0 0 0 4 4 4 0	
	9 8 9 6 16	0 6 6 6 13 4 0 6 5	33 0 15 12 0 23 36 32 0 14	18 24 0 1 6 8 22 18 24 0	20 26 2 0 0 10 24 20 26 2	13 19 19 0 26 10 13 19 18	23 29 5 6 10 0 27 23 29 5 6		0 2 2 9 0 9 9	0 0 0 0 0 0 0 9	40000455050	0 0 0 0 0 0 0 0 0 0 0	440004 44400	0 2 2 4 0 4 0 0 2 2 4	4 0 0 0 9 4 9 9 9	0
	9 8 9 0 16	0 6 8 13 4 0 6 8	33 0 15 12 0 23 36 32 0 14 12 00	18 24 0 1 0 8 22 18 24 0 1	20 26 2 0 0 10 24 20 26 2 0	13 19 0 26 10 13 19 19 19	23 29 5 6 10 0 27 23 29 5 6		0 2 2 4 0 0 2 2 4	0 0 0 9 9 0 0 0 9 0	4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	440004 444000	0 2 2 4 0 4 0 0 2 2 4 0	4 0 4 0 0 0 4 4 6 6 6 6	9
	9 8 9 6 16	0 6 6 6 13 4 0 6 5	33 0 15 12 0 23 36 32 0 14	18 24 0 1 0 8 22 18 24 0 1	20 26 2 0 0 10 24 20 26 2	13 19 19 0 26 10 13 19 18	23 29 5 6 10 0 27 23 29 5 6		0 2 2 9 0 9 9	0 0 0 0 0 0 0 9	40000455050	0 0 0 0 0 0 0 0 0 0 0	440004 44400	0 2 2 4 0 4 0 0 2 2 4	4 0 0 0 9 4 9 9 9	9

			943			Barre S	1000	3111						_/_	_/	
P6=	0	4	36	22	24	10	20	P6=	0	0	5	2	4	0	6	1
	3	0	32	18	20	13	23		0	0	5	0	4	0	4	
	9	6	0	24	26	19	29		2	0	0	2	4	2	4	1
	8	5	14	0	2	18	15		2	0	2	0	0	2	0	1 8
	9	6	12	1	0	19	6		4	4	0	0	0	4	4	1
	00	00	P	00	00	0	io	2 4	0	0	0	0	0	0	0	A
	16	13	22	8	10	26	0		4	4	5	0	4	4	0	3
			10		(11)	19)	7	(+-		34	033	335	0			
DJ=	0	4	36	22	24	10	20	PZ	0	0	5	2	4	0	6	1
	3	0	32	18	20	13	23		0	0	5	0	4	0	4	
	19	6	0	24	26	19	29		2	0	0	2	4	2	4	
Somethin su	8	5	14	0	2	18	5	124	2	0	5	0	0	2	0	
	19	6	12	1	0	11	6		4	4	0	0	0	4	4	
	26	23	22	18	20	0	10	93	7	7	7	7	7	. 0	0	
	16	13	22	8	10	26	0		4	4	5	4	4	4	0	
							,		W	1	319					
		The state of	CHEST OF				19 11 10	-					100			

b. [10 points] Use the Print Shortest Path algorithm (slide 48 of the dynamic programming lecture notes) to find the shortest path from vertex v7 to vertex v3 using the matrix P you constructed from the previous step. Show the actions step by step (either trace the algorithm or show the call tree). You can take the slide 51 as an example of the call tree.

26)	Shortest path from V7 to V3
	Shortest path from V7 to V3 The path is V7 -> V4 -> V5 -> V3
	- P7: 9=7, r=3, (ellpath (7,3)
B 3 4 3	
	- (all path (7,5)
2.45.	
	- (all pull (7,4) -> p(q) (v) = =0, there we return.
1 9 16	THE STATE OF THE PERSON NAMED OF THE PERSON
410-0	- Printing V4
2 10 10 10	
100	- (alling path (4,5) -> p(g) (v) == 0, hence we return.
	- Returning from call to gath (7,5)
	Para
	- Printing Vs
	- Calling pate (4,5) -> P(q)(r) == 0, hence we retorn.
	Caulty for (4), 1)
-0-17/14	- Reforming from call to path (73).
	- Complete
1125-14	- Completer V7, V4, V5, V3.

c. [10 points] Analyze the Print Shortest Path algorithm and show that it has a linear-time complexity (input size is the number of vertices in the graph).
 (Hint: You can consider each array access to P[i][j] as a basic operation.)

Solution:	
9	
2 2 c)	_/_/_
	Armay access P[i][j] is the basic operation for the print att shortest path.
	For vertex Vi to any vertex Vj, the path as, in worst case (n-1) edges, where n's
2	the us. of extens vertices in the graph.
	C- 1 (12-12) odas . H., demillar evaluares & thre
£	For each (N-1) edge, the algorithm evaluates the basic operation P (i) Ij) 4 times, which makes the
2	but $\theta(t)$ $\theta(n)$, which is linear in nature.
2	