ONE DAY FACULTY ORIENTATION PROGRAMME

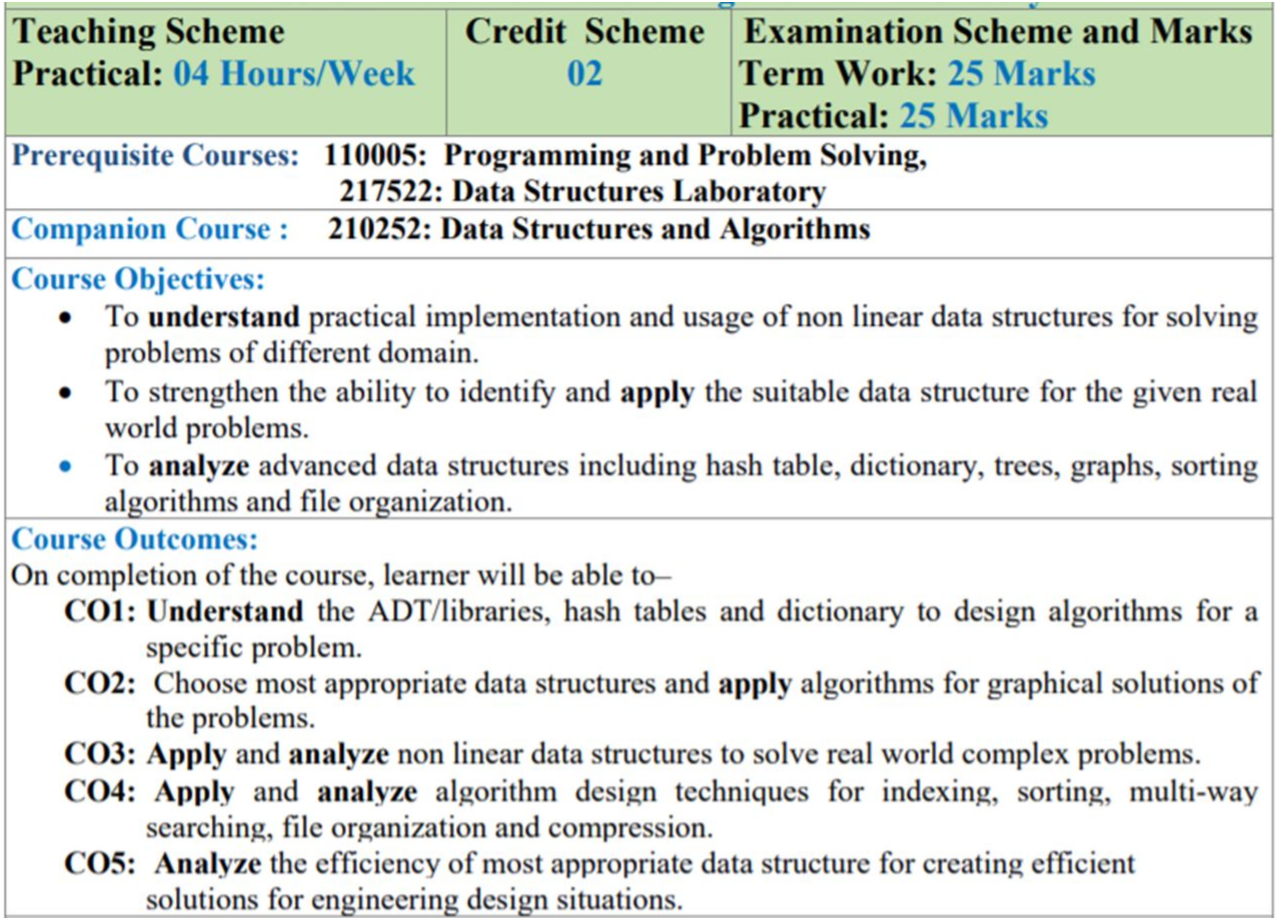
on **Data Structures and Algorithms Lab (217532)** February 16,
in Association with BoS, Computer Engineering, SPPU, Pune

Organized By Department of Artificial Intelligence and Data Science
Dr. D.Y. Patil Institute of Engineering, Management and Research,
Akurdi, Pune

Prepared By

Saudagar Barde

CEO, Pixaflip Technologies



Teaching Scheme Practical: 04 Hours/Week	Credit Scheme 02	Examination Scheme and Marks Term Work: 25 Marks Practical: 25 Marks

Prerequisite Courses: **110005: Programming and Problem Solving,**
217522: Data Structures Laboratory

Companion Course : **210252: Data Structures and Algorithms**

Course Objectives:

- To **understand** practical implementation and usage of non linear data structures for solving problems of different domain.
- To strengthen the ability to identify and **apply** the suitable data structure for the given real world problems.
- To **analyze** advanced data structures including hash table, dictionary, trees, graphs, sorting algorithms and file organization.

Course Outcomes:

On completion of the course, learner will be able to–

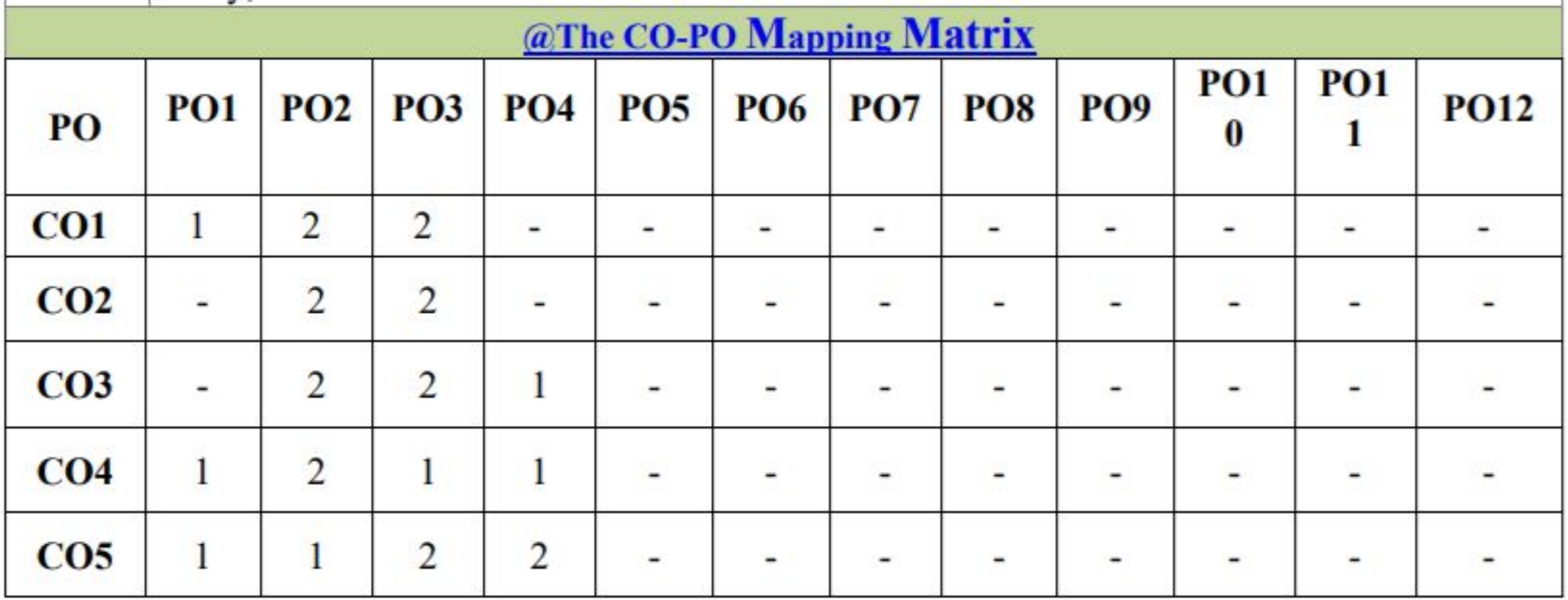
CO1: Understand the ADT/libraries, hash tables and dictionary to design algorithms for a specific problem.

CO2: Choose most appropriate data structures and **apply** algorithms for graphical solutions of the problems.

CO3: Apply and **analyze** non linear data structures to solve real world complex problems.

CO4: Apply and **analyze** algorithm design techniques for indexing, sorting, multi-way searching, file organization and compression.

CO5: Analyze the efficiency of most appropriate data structure for creating efficient solutions for engineering design situations.



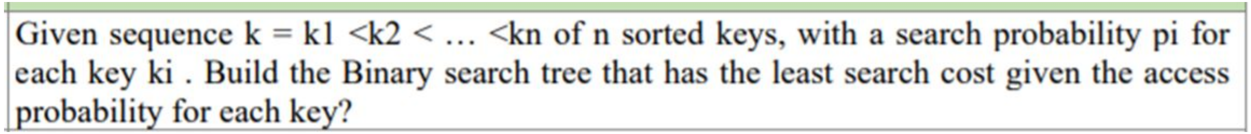
@The CO-PO Mapping Matrix

PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	1	2	2	-	-	-	-	-	-	-	-	-
CO2	-	2	2	-	-	-	-	-	-	-	-	-
CO3	-	2	2	1	-	-	-	-	-	-	-	-
CO4	1	2	1	1	-	-	-	-	-	-	-	-
CO5	1	1	2	2	-	-	-	-	-	-	-	-

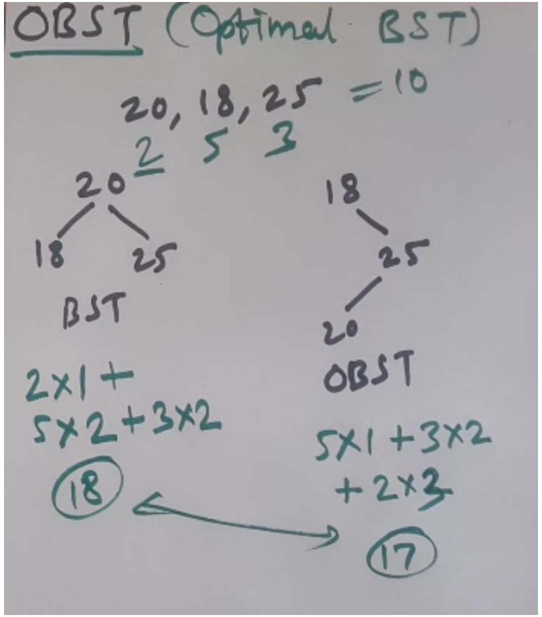
Flipped Classroom

- A flipped classroom is an instructional strategy and a type of blended learning, which aims to increase student engagement and learning by having pupils complete readings at home and work on live problem-solving during class time.
- Prerequisite: Content Delivery Platform like YouTube, etc..

D-18



Given sequence k = k1 <k2 < … <kn of n sorted keys, with a search probability pi for each key ki . Build the Binary search tree that has the least search cost given the access probability for each key?

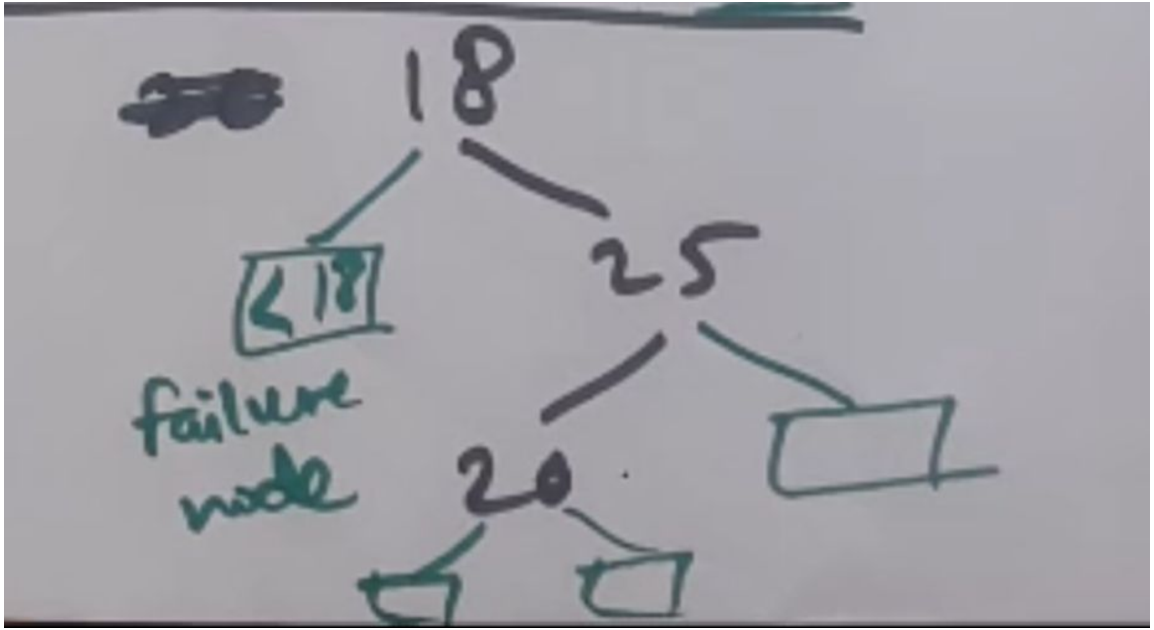


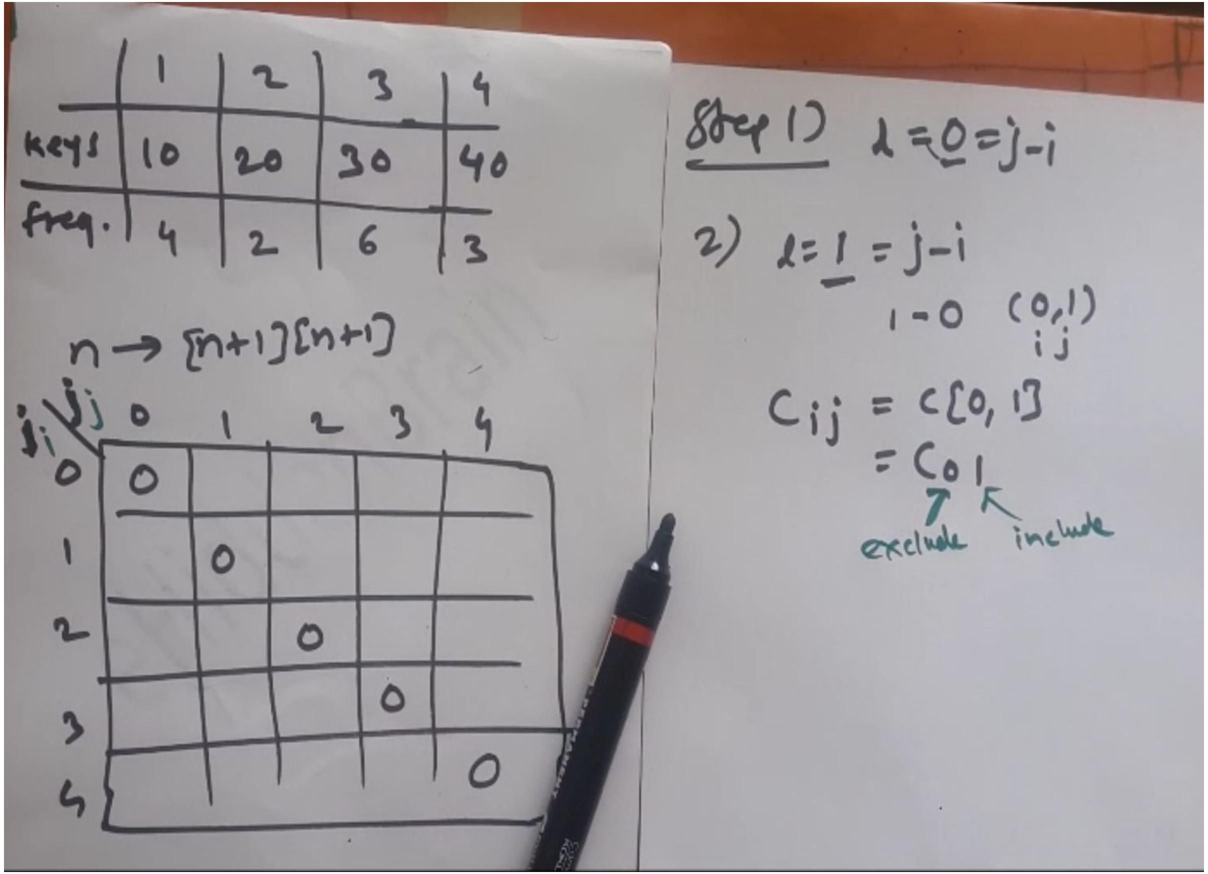
$2\times1 + 5\times2 + 3\times2$

18

$5\times1 + 3\times2 + 2\times3$

17


18
< 18
failure node
25
20



	1	2	3	4
keys	10	20	30	40
freq.	4	2	6	3

$n \rightarrow [n+1][n+1]$

i \ j	0	1	2	3	4
0	0				
1		0			
2			0		
3				0	
4					0

Step 1) $l = \underline{0} = j-i$

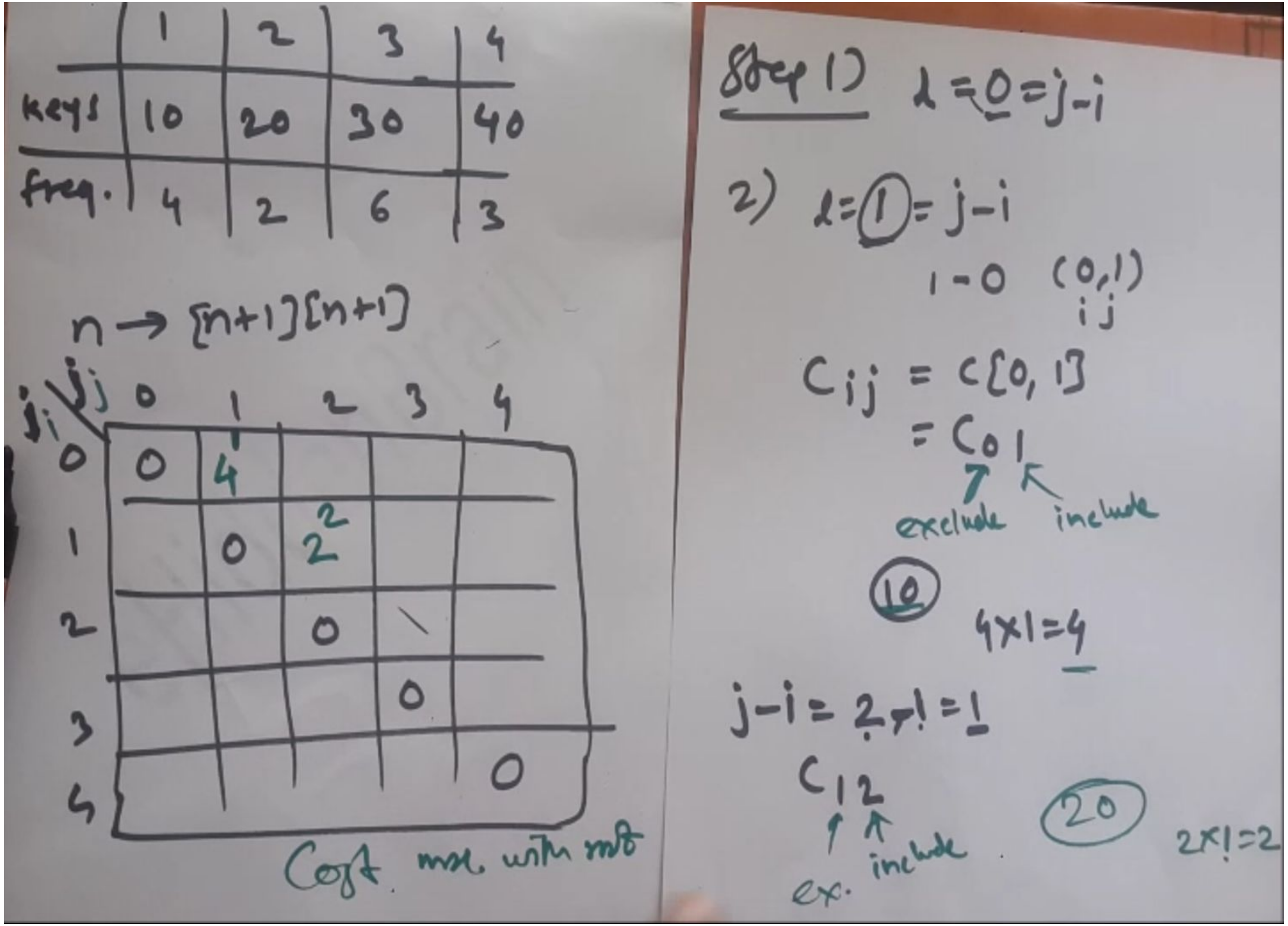
2) $l = \underline{1} = j-i$

$1-0 \quad \underset{i\ \ j}{(0,1)}$

$C_{ij} = c[0,1]$

$= C_{01}$

exclude include



	1	2	3	4
keys	10	20	30	40
freq.	4	2	6	3

n → [n+1][n+1]

i \ j	0	1	2	3	4
0	0	4			
1		0	2 2		
2			0	\	
3				0	
4					0

Cost mat. with mst

Step 1) $l = 0 = j-i$

2) $l = \boxed{1} = j-i$

$1-0 \quad (0,1)$ i j

$C_{ij} = C[0,1]$

$= C_{01}$

exclude include

(10) $4\times1=4$

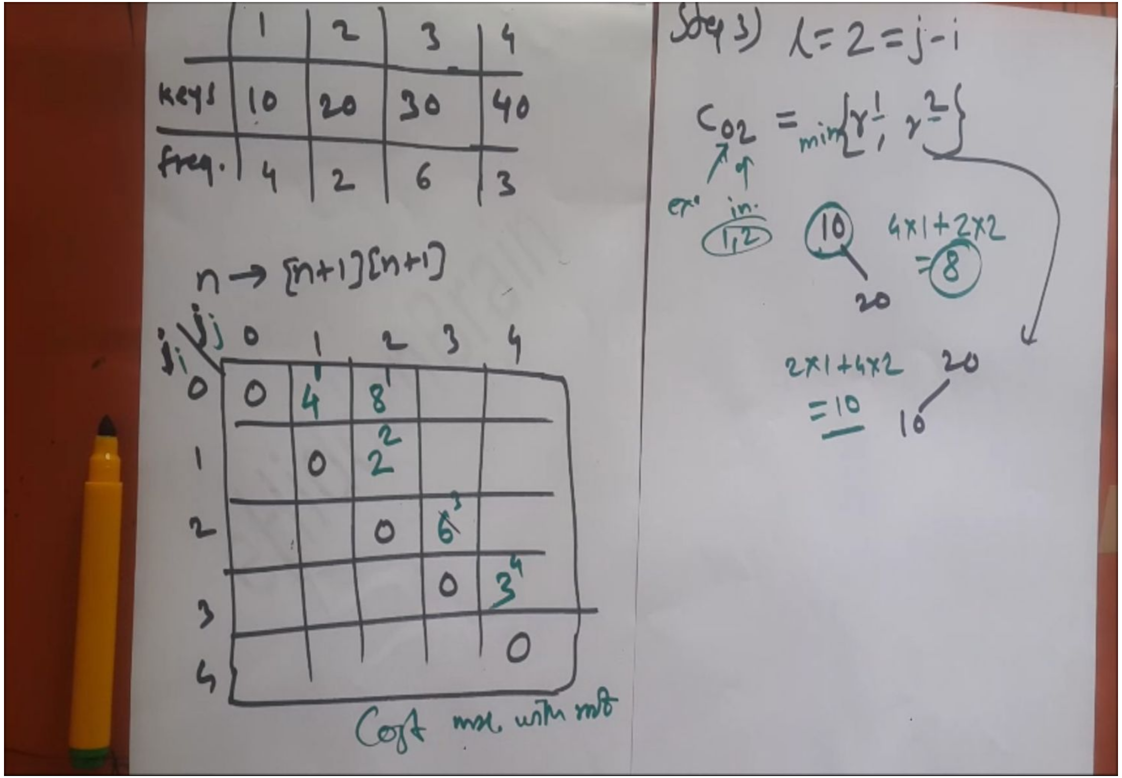
$j-i = 2-1 = 1$

C_{12}

ex. include

(20) $2\times1=2$

Filling the cost matrix repetitively without DP



	1	2	3	4
keys	10	20	30	40
freq.	4	2	6	3

$n \to [n+1][n+1]$

$$\begin{array}{c|ccccc} i \backslash j & 0 & 1 & 2 & 3 & 4 \\ \hline 0 & 0 & 4^{1} & 8^{1} & & \\ 1 & & 0 & 2^{2} & & \\ 2 & & & 0 & 6^{3} & \\ 3 & & & & 0 & 3^{4} \\ 4 & & & & & 0 \end{array}$$

Cost mx. with rt

Step 3) $l = 2 = j - i$

$c_{02} = \min\{r^{1}, r^{2}\}$

ex. in. (1,2)

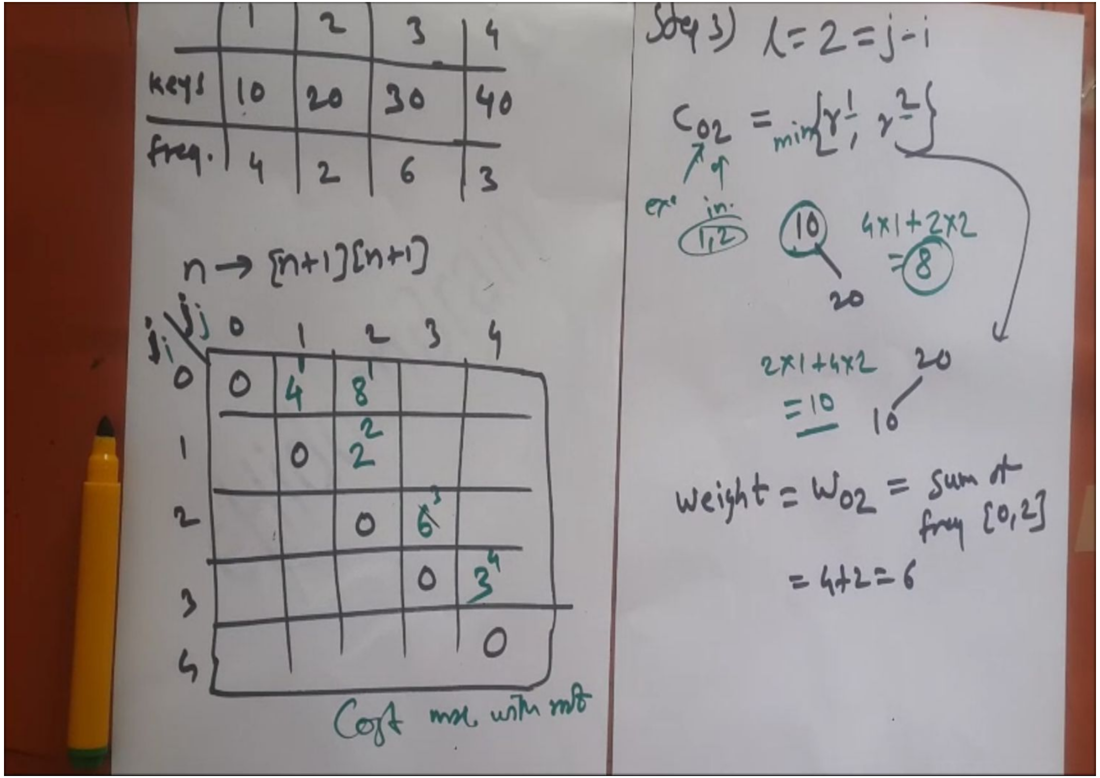
(10) — 20

$4\times1+2\times2 = \boxed{8}$

$2\times1+4\times2 = \underline{10}$

20 — 10

Filling the cost matrix with DP



	1	2	3	4
keys	10	20	30	40
freq.	4	2	6	3

n → [n+1][n+1]

$$\begin{array}{c|ccccc} i \backslash j & 0 & 1 & 2 & 3 & 4 \\ \hline 0 & 0 & 4^{1} & 8^{1} & & \\ 1 & & 0 & 2^{2} & & \\ 2 & & & 0 & 6^{3} & \\ 3 & & & & 0 & 3^{4} \\ 4 & & & & & 0 \end{array}$$

Cost mx. with rt

Step 3) $l = 2 = j - i$

$c_{02} = \min\{r^{1}, r^{2}\}$

ex. in. (1,2)

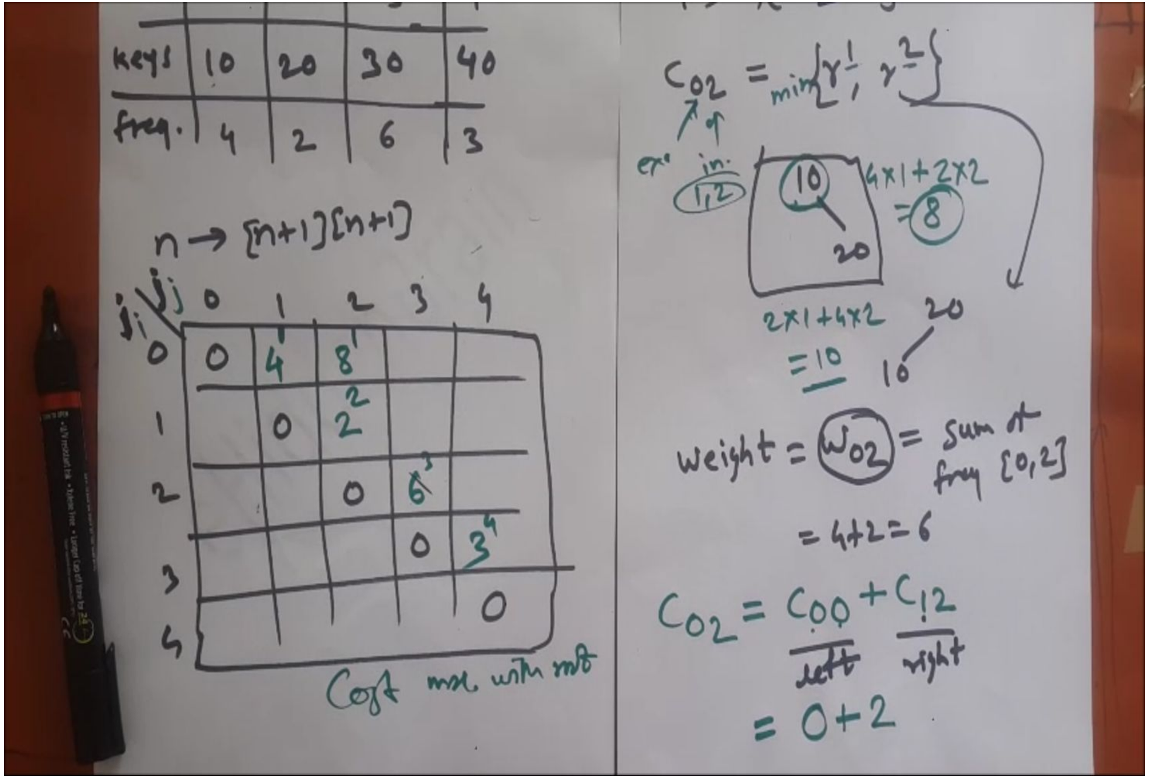
$4\times1 + 2\times2 = 8$

$2\times1 + 4\times2 = 10$

weight $= W_{02} =$ sum of freq [0,2]

$= 4+2 = 6$

Cost of C02 when key 10 is root



keys	10	20	30	40
freq.	4	2	6	3

$n \rightarrow [n+1][n+1]$

i \ j	0	1	2	3	4
0	0	4^1	8^1		
1		0	2^2		
2			0	6^3	
3				0	3^4
4					0

Cost mx. with root

$C_{02} = \min\{r^{1}, r^{2}\}$

ex. in. (1,2)

$4\times1 + 2\times2 = \boxed{8}$

$2\times1 + 4\times2 = 10$

weight = $\boxed{W_{02}}$ = sum of freq [0,2]

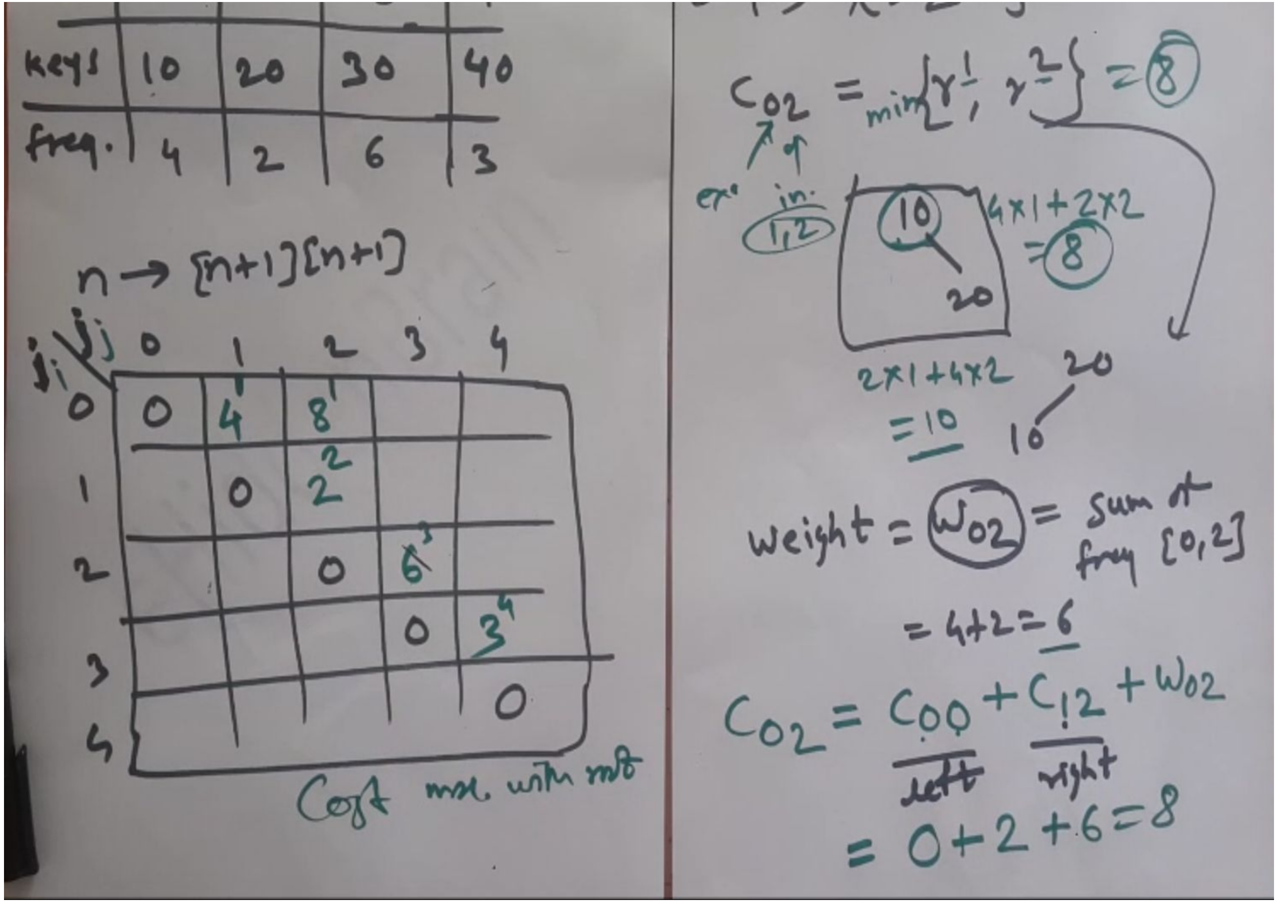
$= 4+2 = 6$

$C_{02} = \underbrace{C_{00}}_{left} + \underbrace{C_{12}}_{right}$

$= 0 + 2$

We got C_{02} as 2, which is wrong

- That means something needs to be added
- There comes W_{02}



keys	10	20	30	40
freq.	4	2	6	3

n → [n+1][n+1]

$$\begin{array}{c|ccccc} i \backslash j & 0 & 1 & 2 & 3 & 4 \\ \hline 0 & 0 & 4^1 & 8^1 & & \\ 1 & & 0 & 2^2 & & \\ 2 & & & 0 & 6^3 & \\ 3 & & & & 0 & 3^4 \\ 4 & & & & & 0 \end{array}$$

Cost mx. with root

$$C_{02} = \min\{ r^1, r^2 \} = \boxed{8}$$

ex. in. $\boxed{1,2}$

$4\times1+2\times2 = \boxed{8}$

$2\times1+4\times2 = \underline{10}$

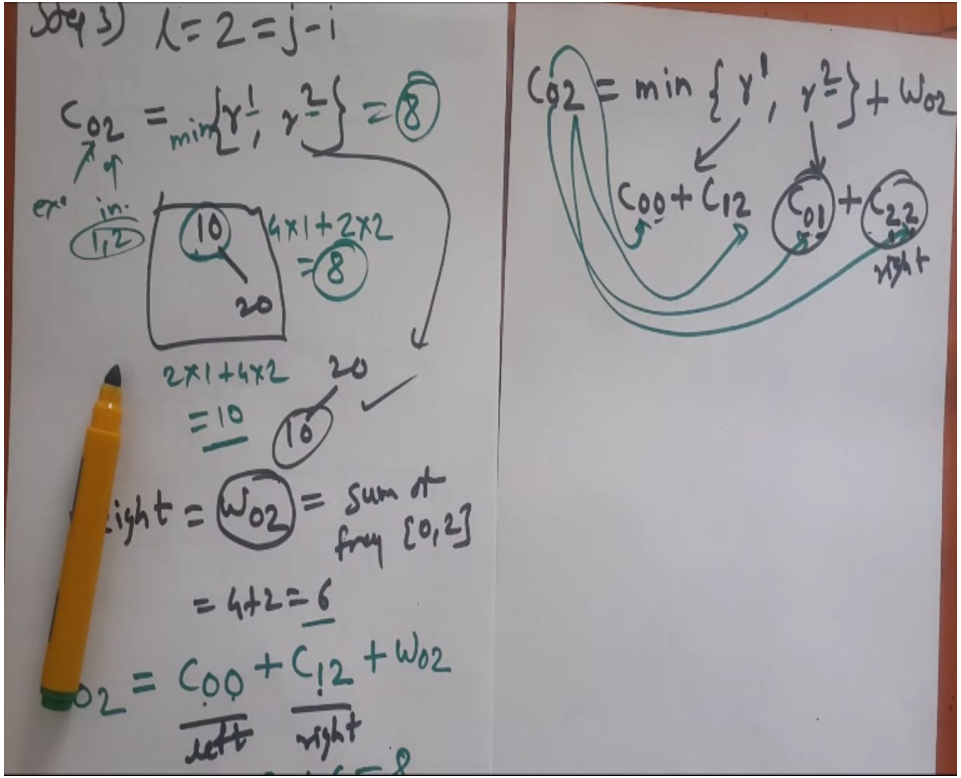
weight = $\boxed{W_{02}}$ = sum of freq [0,2]

$= 4+2 = \underline{6}$

$$C_{02} = \underbrace{C_{00}}_{\text{left}} + \underbrace{C_{12}}_{\text{right}} + W_{02}$$

$= 0+2+6=8$

Making Generic Formula



Step 3) $l = 2 = j - i$

$C_{02} = \min\{r^1, r^2\} = \boxed{8}$

ex. in. 1,2

10

20

$4 \times 1 + 2 \times 2 = \boxed{8}$

$2 \times 1 + 4 \times 2 = 10$

20

10 ✓

ight = W_{02} = sum of freq [0,2]

$= 4 + 2 = 6$

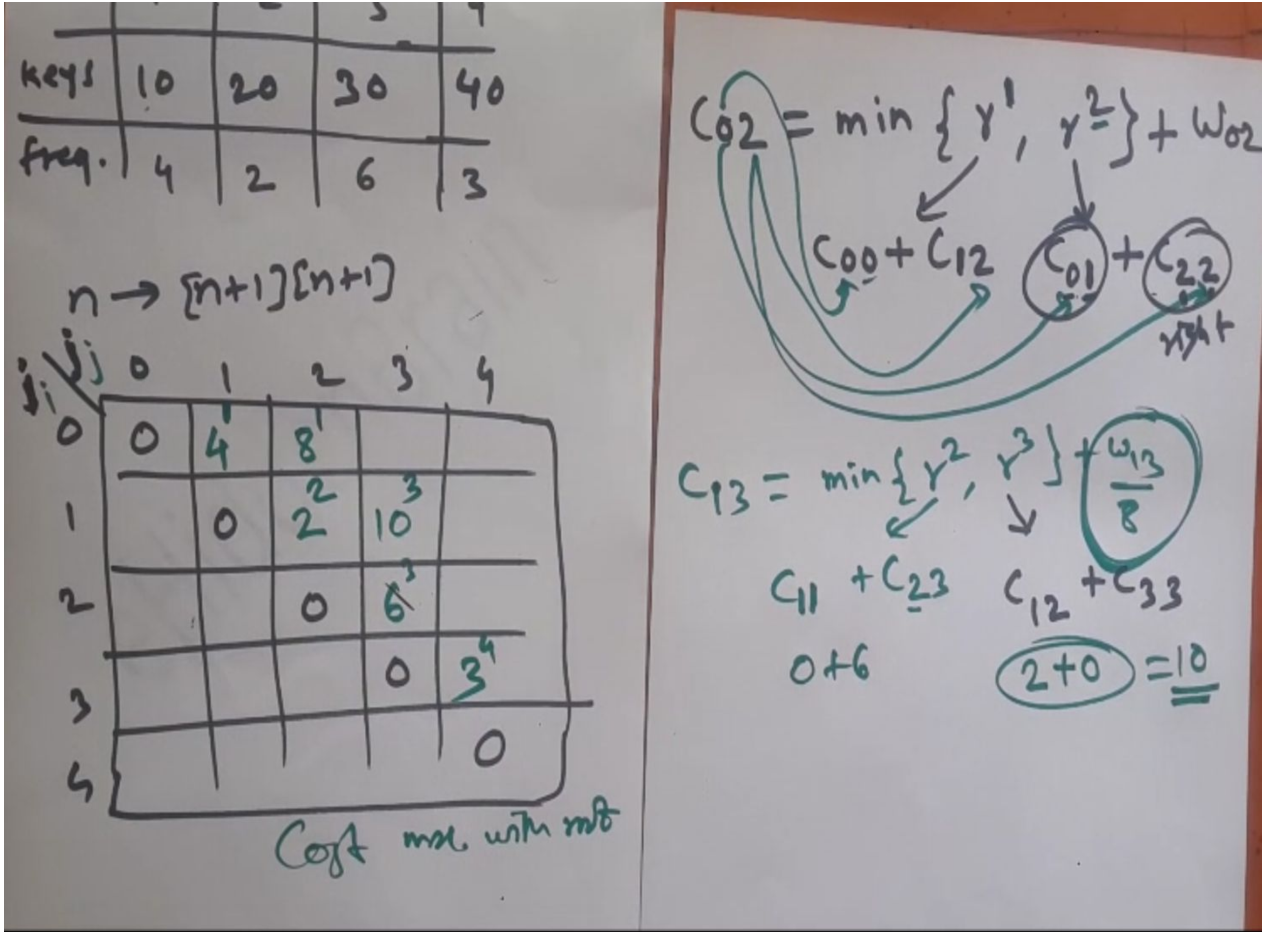
$C_{02} = \underbrace{C_{00}}_{\text{left}} + \underbrace{C_{12}}_{\text{right}} + W_{02}$

$C_{02} = \min\{r^1, r^2\} + W_{02}$

$C_{00} + C_{12}$

$C_{01} + C_{22}$

right



	1	2	3	4
keys	10	20	30	40
freq.	4	2	6	3

$n \rightarrow [n+1][n+1]$

$$\begin{array}{c|ccccc} i \backslash j & 0 & 1 & 2 & 3 & 4 \\ \hline 0 & 0 & 4^1 & 8^1 & & \\ 1 & & 0 & 2^2 & 10^3 & \\ 2 & & & 0 & 6^3 & \\ 3 & & & & 0 & 3^4 \\ 4 & & & & & 0 \end{array}$$

Cost mx. with rt

$C_{02} = \min\{r^1, r^2\} + W_{02}$

$r^1 \rightarrow C_{00} + C_{12}$

$r^2 \rightarrow C_{01} + C_{22}$

right

$C_{13} = \min\{r^2, r^3\} + \frac{w_{13}}{8}$

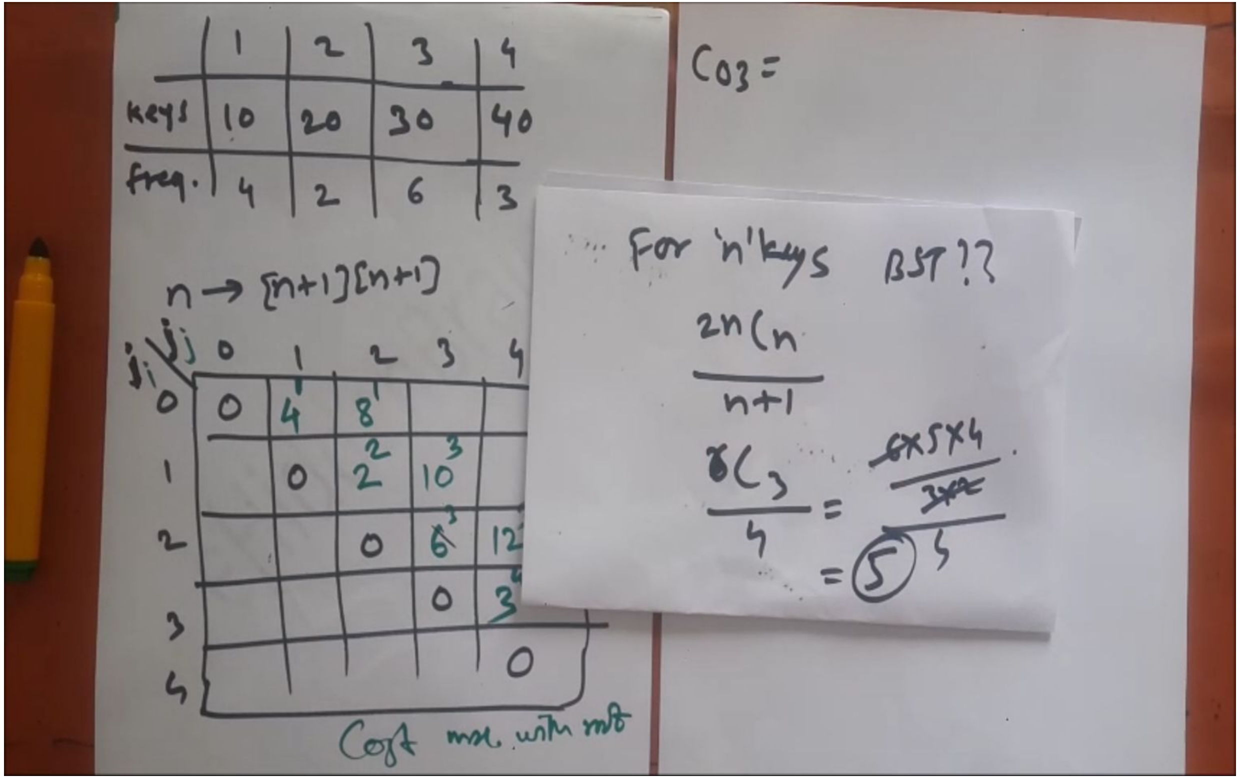
$r^2 \rightarrow C_{11} + C_{23}$

$0+6$

$r^3 \rightarrow C_{12} + C_{33}$

$2+0 = \underline{\underline{10}}$

Why DP is useful?



	1	2	3	4
keys	10	20	30	40
freq.	4	2	6	3

$n \rightarrow [n+1][n+1]$

i \ j	0	1	2	3	4
0	0	4^1	8^1		
1		0	2^2	10^3	
2			0	6^3	12
3				0	3^4
4					0

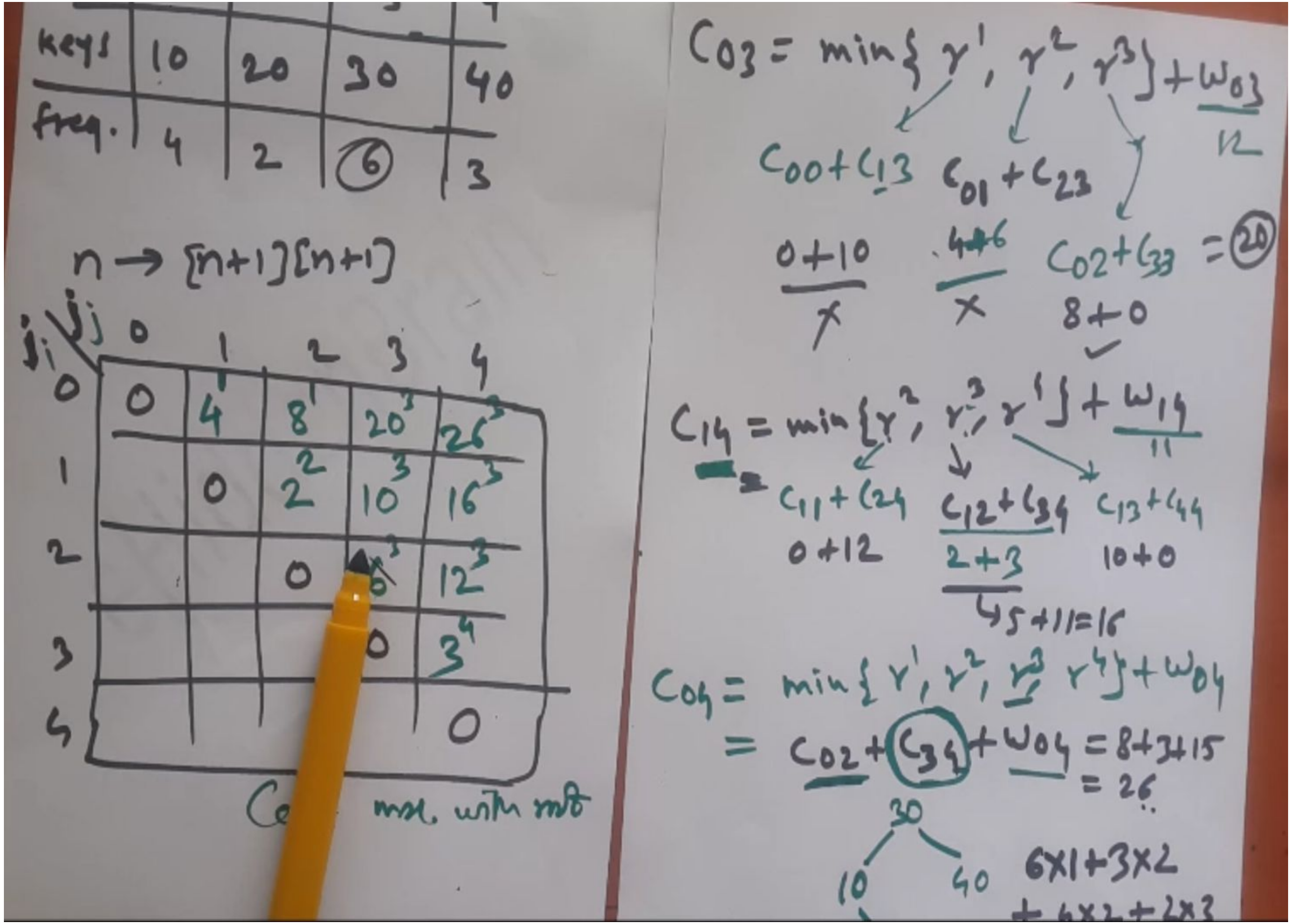
Cost mx. with root

$C_{03} =$

For 'n' keys BST ??

$$\frac{{}^{2n}C_n}{n+1}$$

$$\frac{{}^{6}C_3}{4} = \frac{\frac{\cancel{6}\times5\times4}{\cancel{3\times2}}}{4} = \boxed{5}$$



keys	10	20	30	40
freq.	4	2	$\textcircled{6}$	3

$n \rightarrow [n+1][n+1]$

$j_i \backslash j_j$	0	1	2	3	4
0	0	4^1	8^1	20^3	26^3
1		0	2^2	10^3	16^3
2			0	6^3	12^3
3				0	3^4
4					0

Co[illegible] mxt. with rmt

$$C_{03} = \min\{r^1, r^2, r^3\} + \underline{w_{03}}$$

$\underline{w_{03}} = 12$

$C_{00}+C_{13}$ $\quad$ $C_{01}+C_{23}$ $\quad$ $C_{02}+C_{33} = \boxed{20}$

$\underline{0+10}$ ✗ $\quad$ $\underline{4+6}$ ✗ $\quad$ $8+0$ ✓

$$C_{14} = \min\{r^2, r^3, r^1\} + \underline{w_{14}}$$

$\underline{w_{14}} = 11$

$= C_{11}+C_{24}$ $\quad$ $\underline{C_{12}+C_{34}}$ $\quad$ $C_{13}+C_{44}$

$0+12$ $\quad$ $\underline{2+3}$ $\quad$ $10+0$

$\hookrightarrow 5+11=16$

$$C_{04} = \min\{r^1, r^2, \underline{r^3}\ r^4\} + w_{04}$$

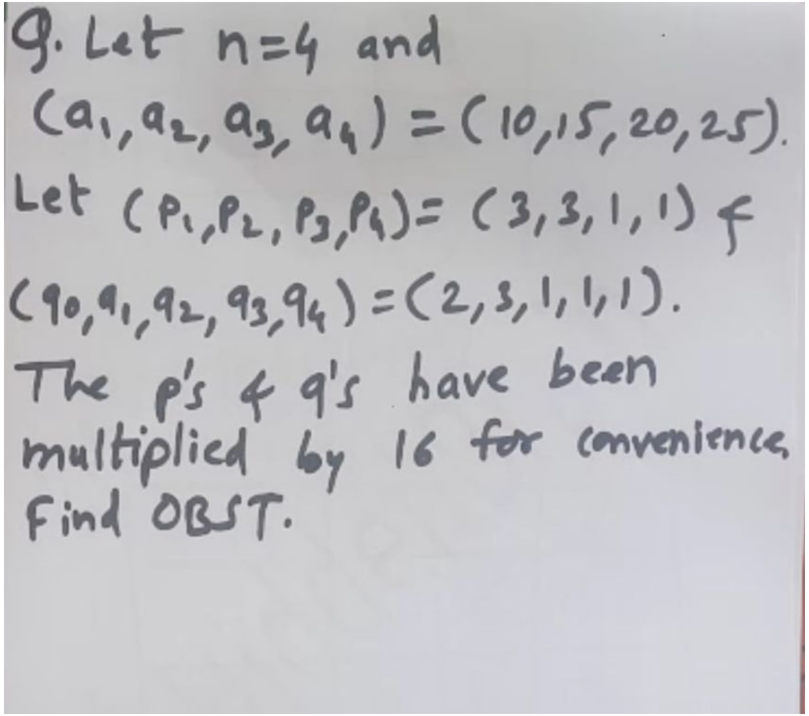
$$= \underline{C_{02}} + C_{34} + \underline{W_{04}} = 8+3+15$$

$$= 26$$

$6\times1+3\times2$

$+ 4\times2 + 2\times3$

Example



Q. Let $n=4$ and

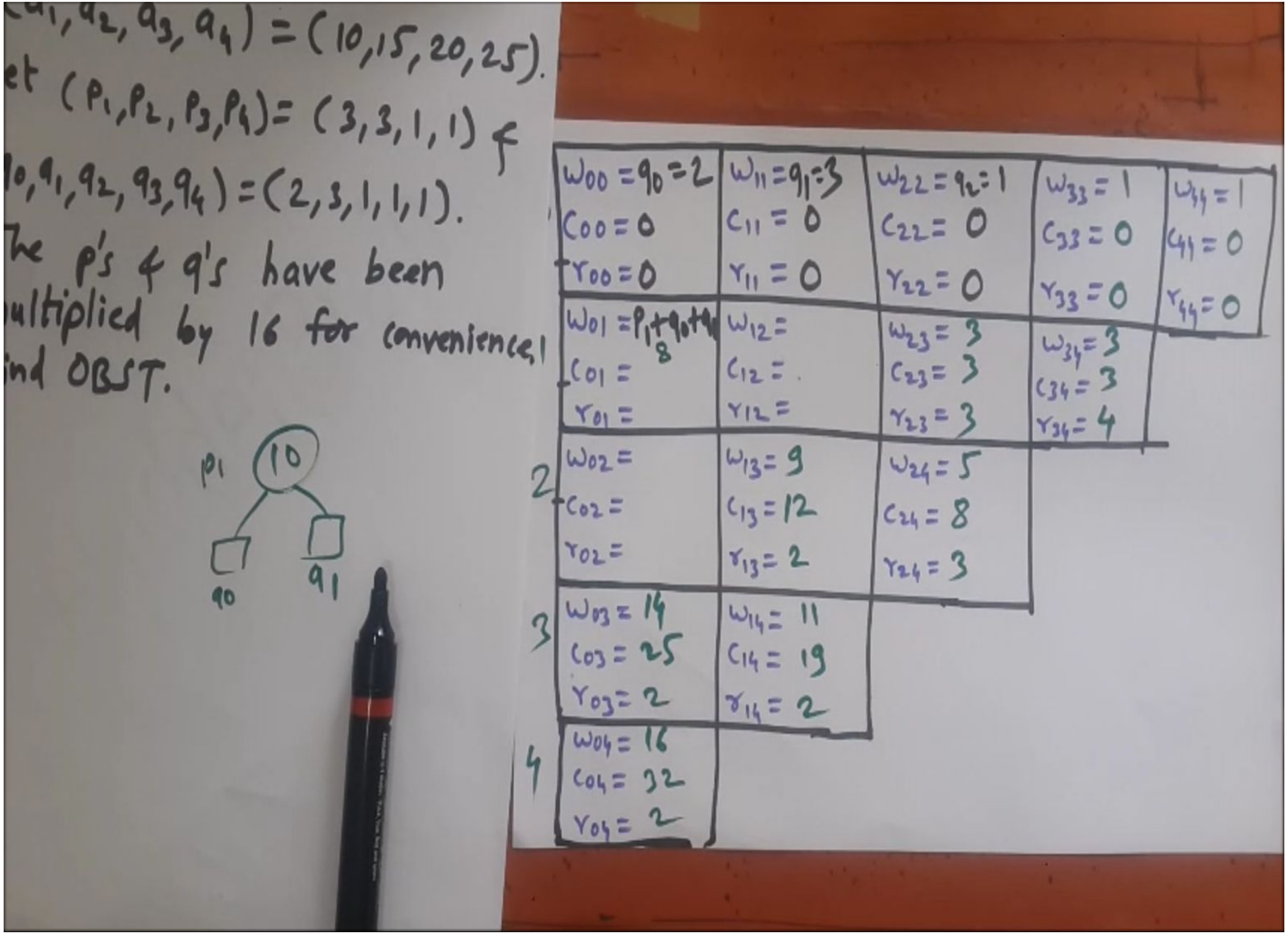
$(a_1, a_2, a_3, a_4) = (10, 15, 20, 25)$.

Let $(p_1, p_2, p_3, p_4) = (3, 3, 1, 1)$ &

$(q_0, q_1, q_2, q_3, q_4) = (2, 3, 1, 1, 1)$.

The p's & q's have been multiplied by 16 for convenience.

Find OBST.



$(a_1, a_2, a_3, a_4) = (10, 15, 20, 25)$.

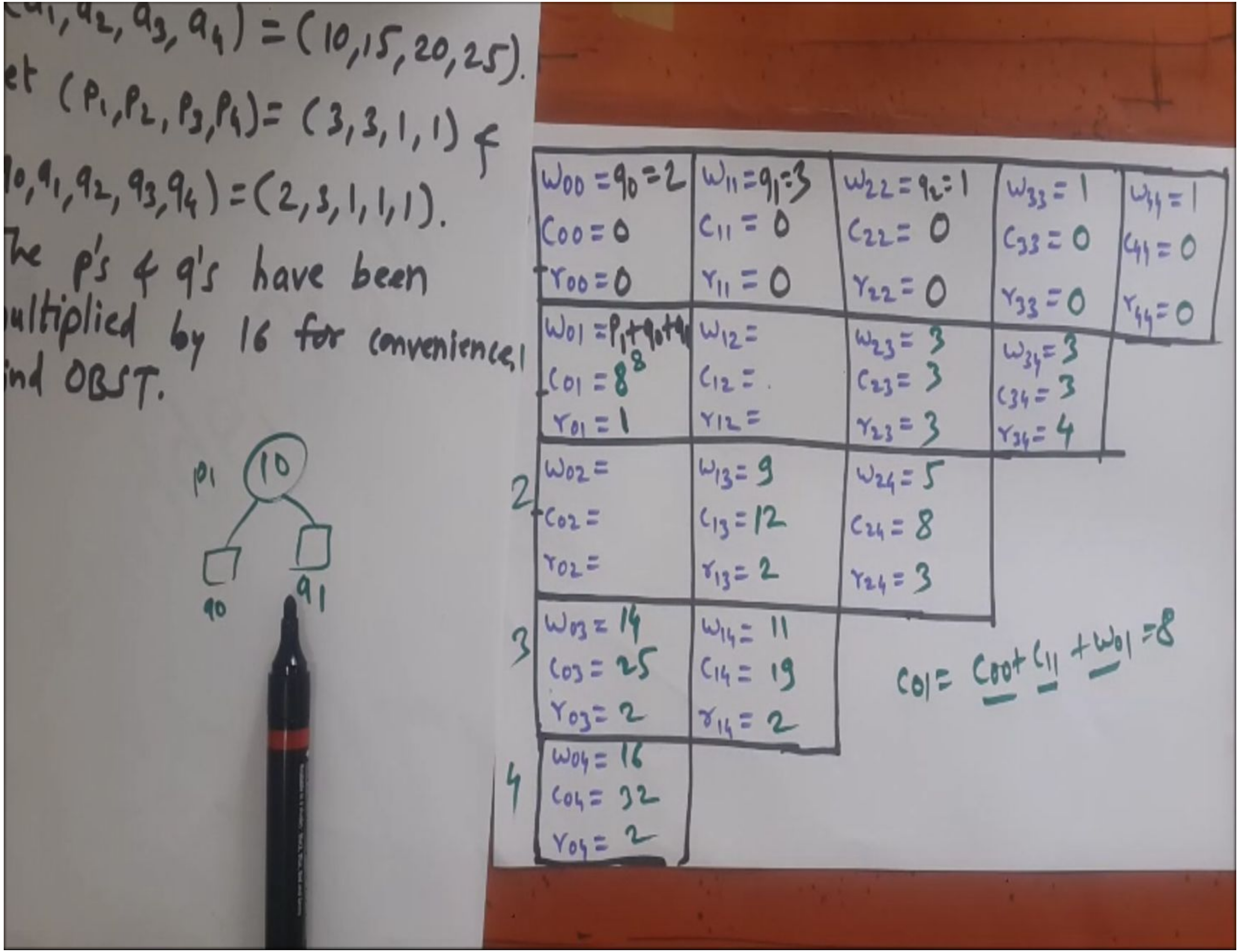
et $(p_1, p_2, p_3, p_4) = (3, 3, 1, 1)$ &

$q_0, q_1, q_2, q_3, q_4) = (2, 3, 1, 1, 1)$.

he p's & q's have been ultiplied by 16 for convenience.

ind OBST.

	$w_{00} = q_0 = 2$ $c_{00} = 0$ $r_{00} = 0$	$w_{11} = q_1 = 3$ $c_{11} = 0$ $r_{11} = 0$	$w_{22} = q_2 = 1$ $c_{22} = 0$ $r_{22} = 0$	$w_{33} = 1$ $c_{33} = 0$ $r_{33} = 0$	$w_{44} = 1$ $c_{44} = 0$ $r_{44} = 0$
1	$w_{01} = p_1 + q_0 + q_1$ 8 $c_{01} =$ $r_{01} =$	$w_{12} =$ $c_{12} =$ $r_{12} =$	$w_{23} = 3$ $c_{23} = 3$ $r_{23} = 3$	$w_{34} = 3$ $c_{34} = 3$ $r_{34} = 4$	
2	$w_{02} =$ $c_{02} =$ $r_{02} =$	$w_{13} = 9$ $c_{13} = 12$ $r_{13} = 2$	$w_{24} = 5$ $c_{24} = 8$ $r_{24} = 3$		
3	$w_{03} = 14$ $c_{03} = 25$ $r_{03} = 2$	$w_{14} = 11$ $c_{14} = 19$ $r_{14} = 2$			
4	$w_{04} = 16$ $c_{04} = 32$ $r_{04} = 2$				



$(a_1, a_2, a_3, a_4) = (10, 15, 20, 25)$.

et $(p_1, p_2, p_3, p_4) = (3, 3, 1, 1)$ &

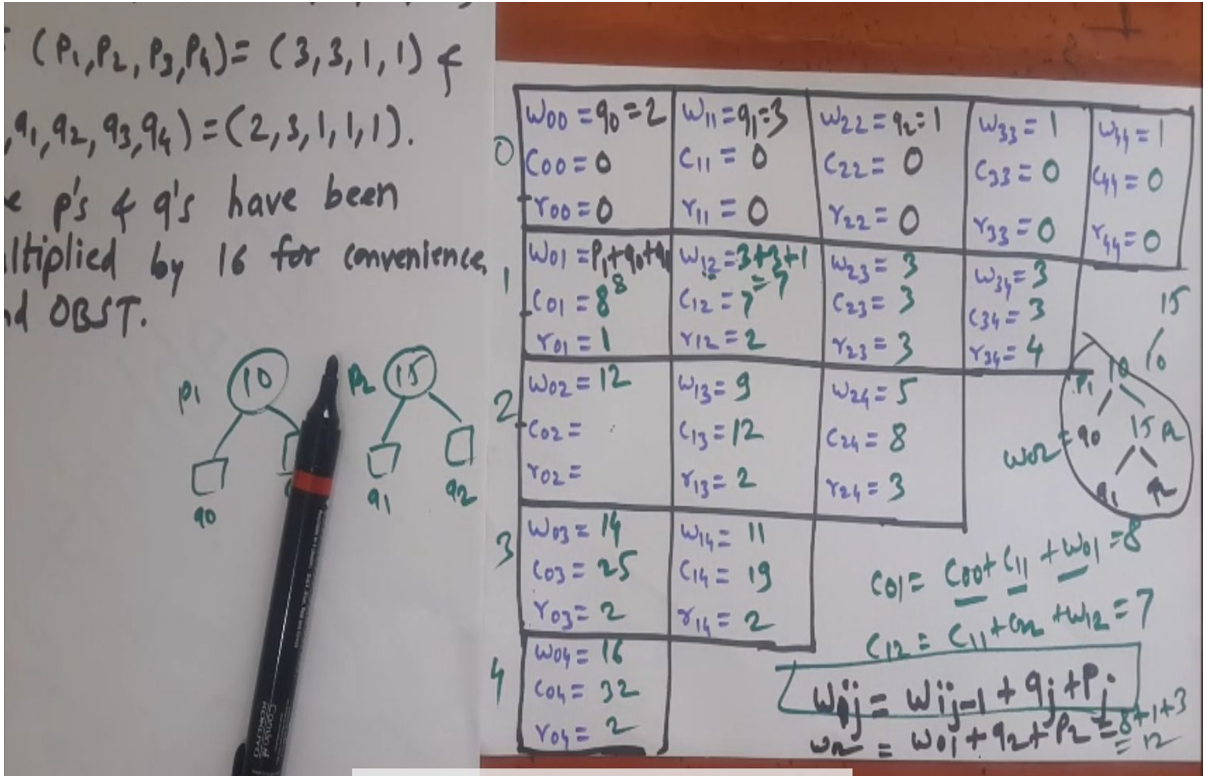
$q_0, q_1, q_2, q_3, q_4) = (2, 3, 1, 1, 1)$.

he p's & q's have been
ultiplied by 16 for convenience,
ind OBST.

	$w_{00} = q_0 = 2$ $c_{00} = 0$ $r_{00} = 0$	$w_{11} = q_1 = 3$ $c_{11} = 0$ $r_{11} = 0$	$w_{22} = q_2 = 1$ $c_{22} = 0$ $r_{22} = 0$	$w_{33} = 1$ $c_{33} = 0$ $r_{33} = 0$	$w_{44} = 1$ $c_{44} = 0$ $r_{44} = 0$
	$w_{01} = p_1 + q_0 + q_1$ $c_{01} = 8^{8}$ $r_{01} = 1$	$w_{12} =$ $c_{12} =$ $r_{12} =$	$w_{23} = 3$ $c_{23} = 3$ $r_{23} = 3$	$w_{34} = 3$ $c_{34} = 3$ $r_{34} = 4$	
2	$w_{02} =$ $c_{02} =$ $r_{02} =$	$w_{13} = 9$ $c_{13} = 12$ $r_{13} = 2$	$w_{24} = 5$ $c_{24} = 8$ $r_{24} = 3$		
3	$w_{03} = 14$ $c_{03} = 25$ $r_{03} = 2$	$w_{14} = 11$ $c_{14} = 19$ $r_{14} = 2$			
4	$w_{04} = 16$ $c_{04} = 32$ $r_{04} = 2$				

$c_{01} = \underline{c_{00}} + \underline{c_{11}} + \underline{w_{01}} = 8$

Using DP for W_{ij} too



$(p_1,p_2,p_3,p_4)=(3,3,1,1)$ &

$,q_1,q_2,q_3,q_4)=(2,3,1,1,1)$.

e p's & q's have been

ltiplied by 16 for convenience,

d OBST.

p1
10
p2
15
q0
q1
q2

0	$w_{00}=q_0=2$ $c_{00}=0$ $r_{00}=0$	$w_{11}=q_1=3$ $c_{11}=0$ $r_{11}=0$	$w_{22}=q_2=1$ $c_{22}=0$ $r_{22}=0$	$w_{33}=1$ $c_{33}=0$ $r_{33}=0$	$w_{44}=1$ $c_{44}=0$ $r_{44}=0$
1	$w_{01}=p_1+q_0+q_1$ ^{8} $c_{01}=8$ $r_{01}=1$	$w_{12}=3+3+1=7$ $c_{12}=7$ $r_{12}=2$	$w_{23}=3$ $c_{23}=3$ $r_{23}=3$	$w_{34}=3$ $c_{34}=3$ $r_{34}=4$	
2	$w_{02}=12$ $c_{02}=$ $r_{02}=$	$w_{13}=9$ $c_{13}=12$ $r_{13}=2$	$w_{24}=5$ $c_{24}=8$ $r_{24}=3$		
3	$w_{03}=14$ $c_{03}=25$ $r_{03}=2$	$w_{14}=11$ $c_{14}=19$ $r_{14}=2$			
4	$w_{04}=16$ $c_{04}=32$ $r_{04}=2$				

15
10
$w_{02}=$
10 P1
q0
15 P2
q1
q2

$c_{01} = \underline{c_{00}} + \underline{c_{11}} + \underline{w_{01}} = 8$

$c_{12} = c_{11} + c_{22} + w_{12} = 7$

$$\boxed{w_{ij} = w_{ij-1} + q_j + p_j}$$

$w_{02} = w_{01} + q_2 + p_2 = 8+1+3 = 12$

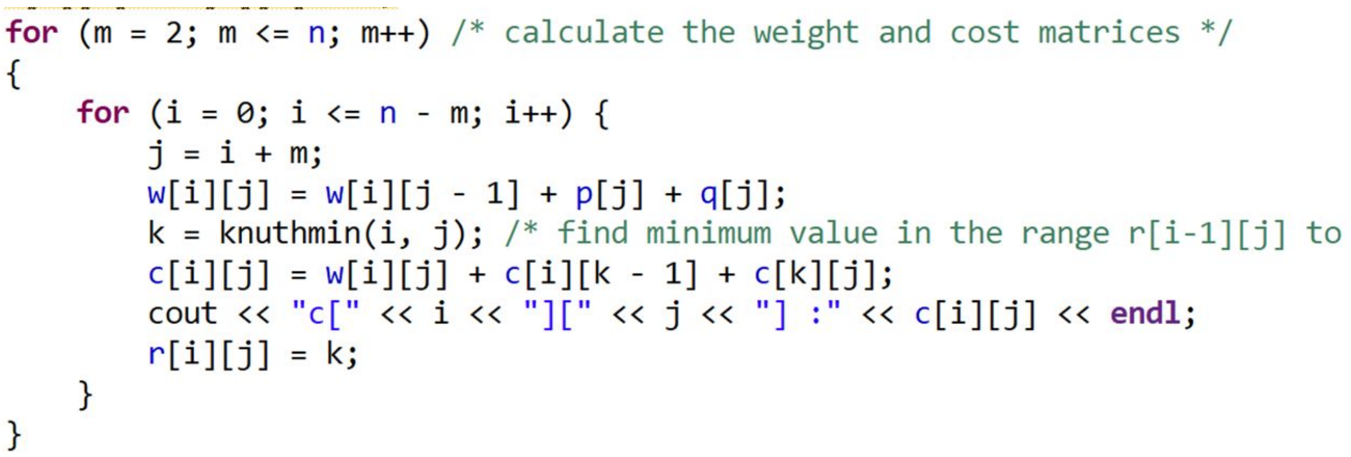
Observations

- 2nd key is considered as root though 1st key is also having same success frequency. Because, failure frequencies attached with 1st key are 2,3 while failure frequencies attached with 2nd key are 3,1. So here also algorithm prefers optimization.

- Dynamic Programming for NP-class Problems.

Time complexity

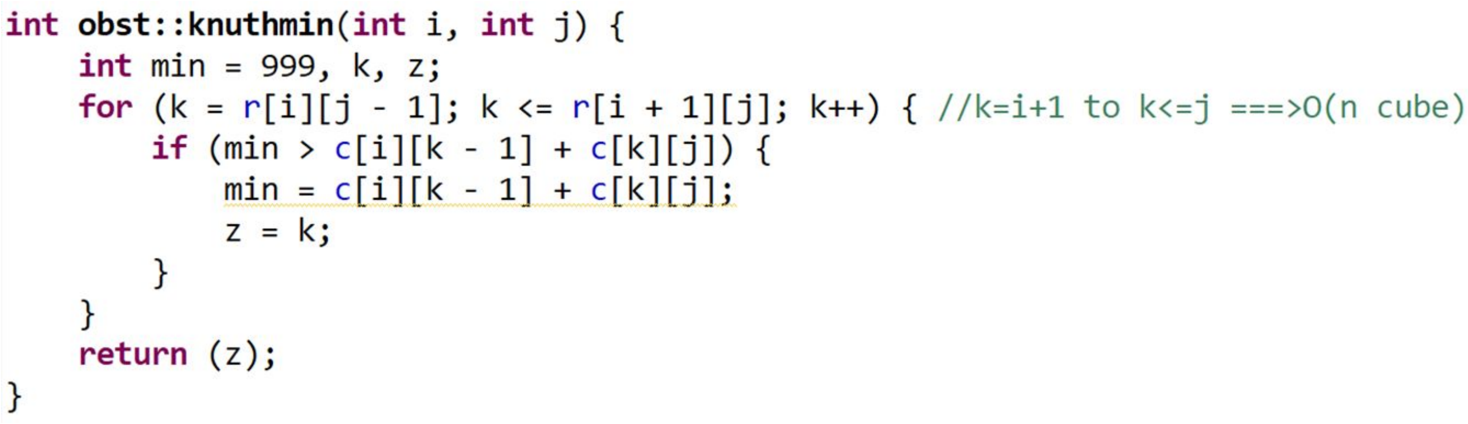
- **$O(n^3)$ where n = keys**



```
for (m = 2; m <= n; m++) /* calculate the weight and cost matrices */
{
    for (i = 0; i <= n - m; i++) {
        j = i + m;
        w[i][j] = w[i][j - 1] + p[j] + q[j];
        k = knuthmin(i, j); /* find minimum value in the range r[i-1][j] to
        c[i][j] = w[i][j] + c[i][k - 1] + c[k][j];
        cout << "c[" << i << "][" << j << "] :" << c[i][j] << endl;
        r[i][j] = k;
    }
}
```

Knuth reduced TC

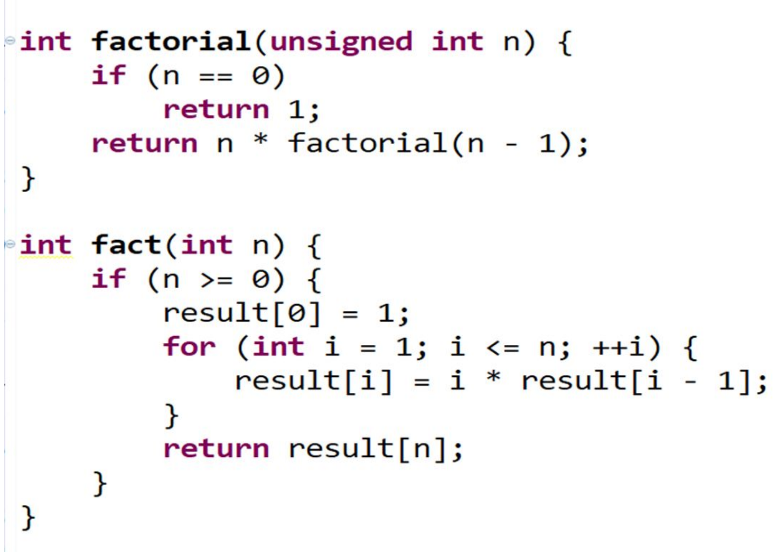
- $O(n^2)$



```
int obst::knuthmin(int i, int j) {
    int min = 999, k, z;
    for (k = r[i][j - 1]; k <= r[i + 1][j]; k++) { //k=i+1 to k<=j ===>O(n cube)
        if (min > c[i][k - 1] + c[k][j]) {
            min = c[i][k - 1] + c[k][j];
            z = k;
        }
    }
    return (z);
}
```

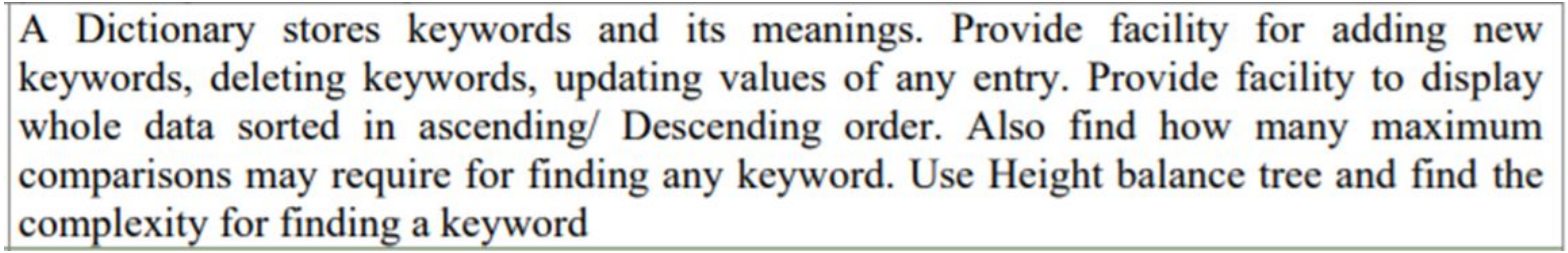
Benefits of DP

- **DP vs Recursion**



```
int factorial(unsigned int n) {
    if (n == 0)
        return 1;
    return n * factorial(n - 1);
}

int fact(int n) {
    if (n >= 0) {
        result[0] = 1;
        for (int i = 1; i <= n; ++i) {
            result[i] = i * result[i - 1];
        }
        return result[n];
    }
}
```

D-19



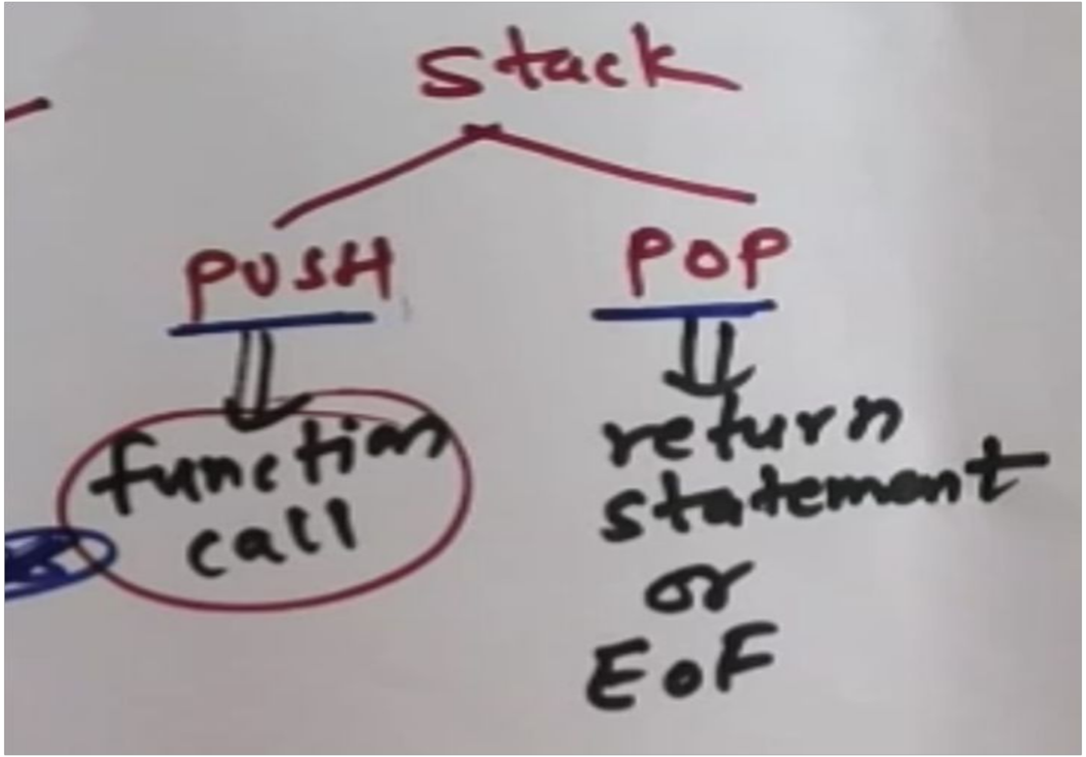
A Dictionary stores keywords and its meanings. Provide facility for adding new keywords, deleting keywords, updating values of any entry. Provide facility to display whole data sorted in ascending/ Descending order. Also find how many maximum comparisons may require for finding any keyword. Use Height balance tree and find the complexity for finding a keyword

AVL Tree with Recursion

Challenge – How to teach?

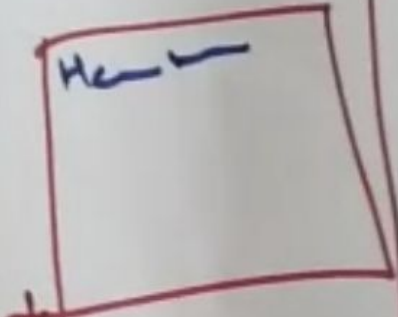
Teach AVL tree 1st with all operations
RR, LL, RL, LR

Teach basic of recursions



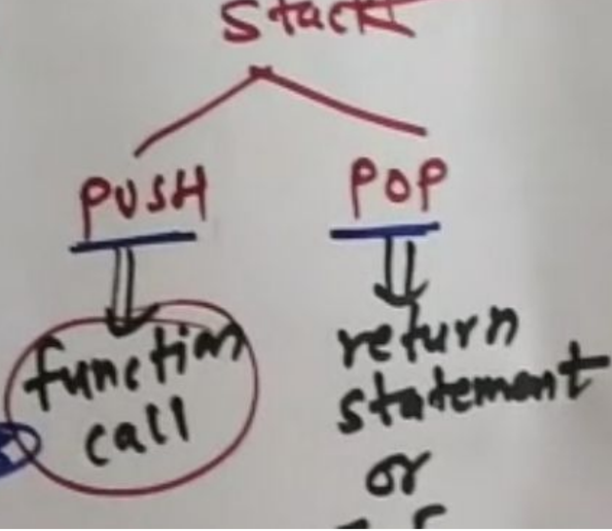
```
10 int main(){
11   printf("Hello World \n");
12   add();
13   printf("Exited");
14   return 0;
15 }
```

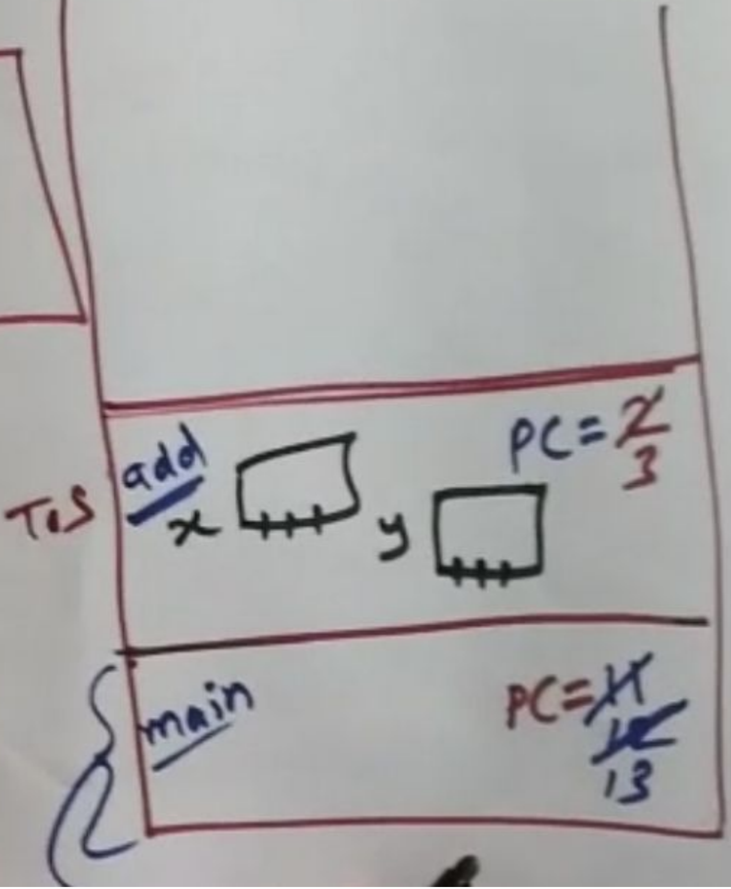
PC = is next line to be executed



syntax ✓

object

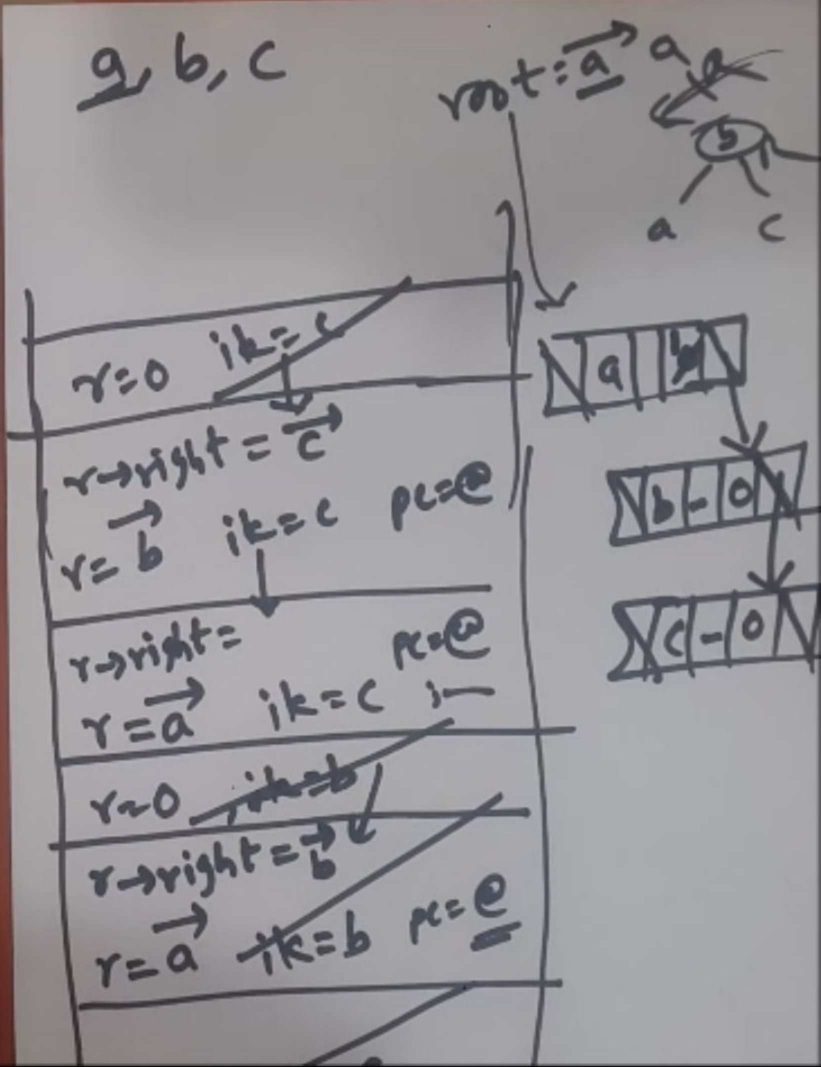




4

6

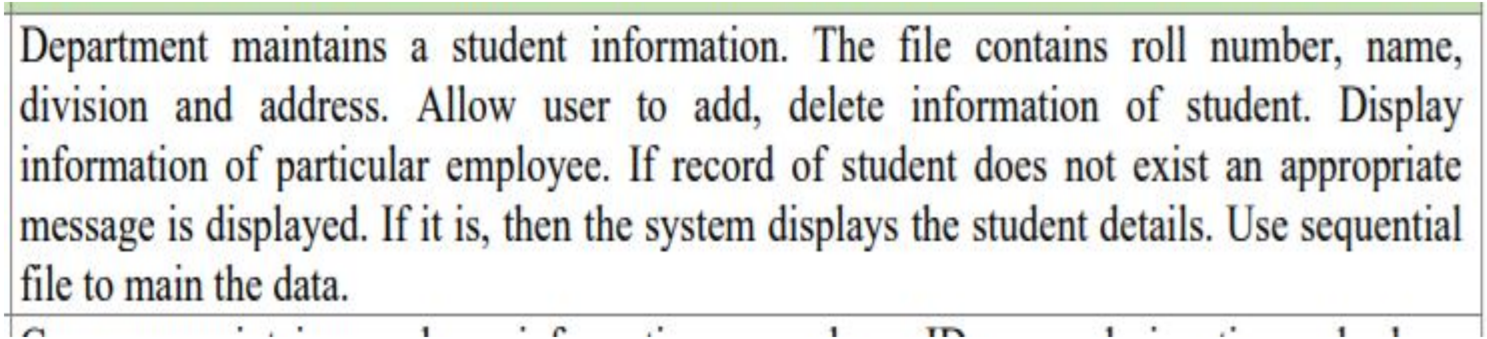
```
    if (lheight > rheight)
        return lheight;
    else
        return rheight;
}

struct node *avldictionary::insertkeyword(struct node *r,
char ik[15],
        char im[15]) {
    if (r == NULL) {
        r = new struct node;
        strcpy(r->keyword, ik); //r's keyword and meaning
        strcpy(r->meaning, im); //updated with values given
//by user
        r->left = r->right = NULL;  //r's both links are
//set to NULL
    } else if (strcmp(ik, r->keyword) > 0) {
        r->right = insertkeyword(r->right, ik, im);

        if (balanceFactor(r) == -2) //BF is -2 then @
//insertion in RightSubTree
            {
            if (strcmp      ->right->keyword) > 0) {
```



Time & Space complexity of Recursive functions

https://stackoverflow.com/questions/13467674/determining-complexity-for-recursive-functions-big-o-notation

F-23

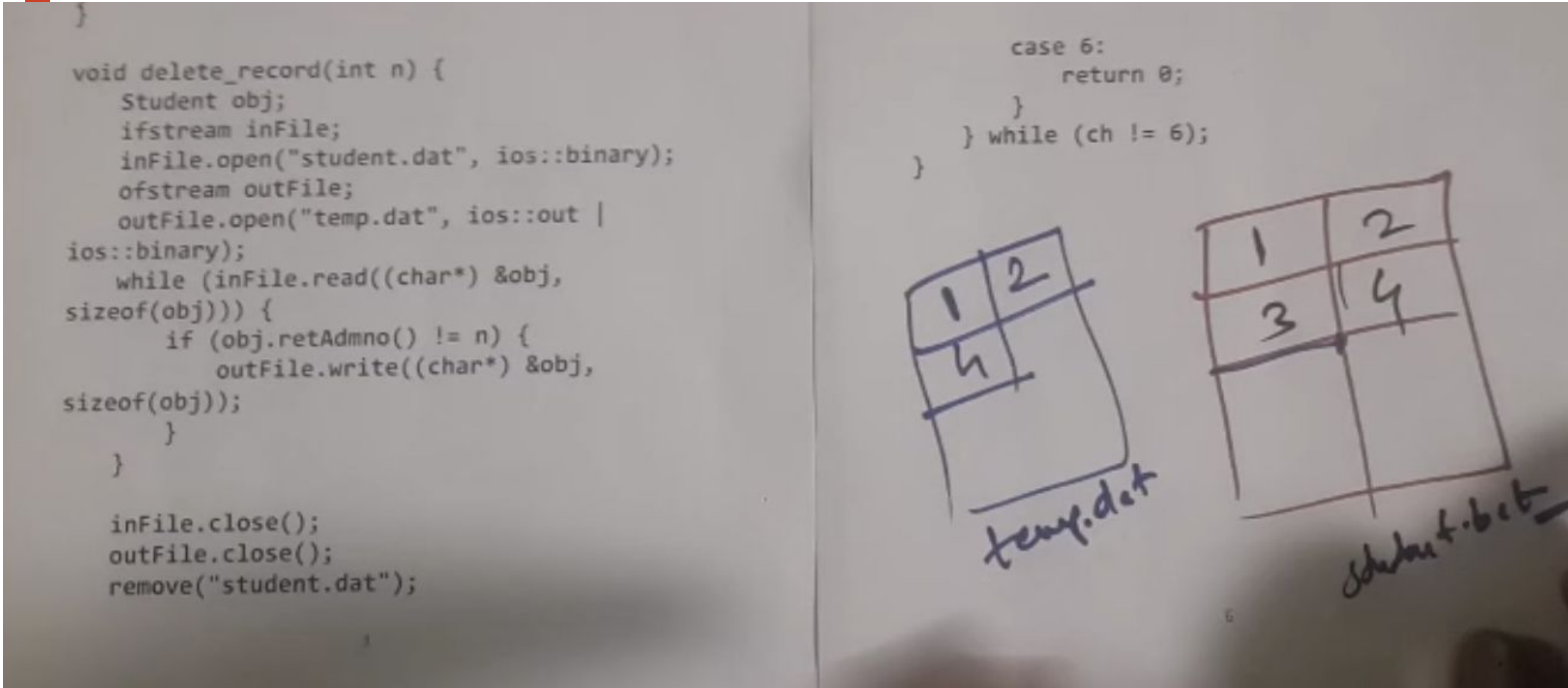


Department maintains a student information. The file contains roll number, name, division and address. Allow user to add, delete information of student. Display information of particular employee. If record of student does not exist an appropriate message is displayed. If it is, then the system displays the student details. Use sequential file to main the data.

ofstream, ifstream, seekp, seekg

Useful for FIFO operations and cheaper device storages

Deletion is costly operation



```
}

void delete_record(int n) {
    Student obj;
    ifstream inFile;
    inFile.open("student.dat", ios::binary);
    ofstream outFile;
    outFile.open("temp.dat", ios::out |
ios::binary);
    while (inFile.read((char*) &obj,
sizeof(obj))) {
        if (obj.retAdmno() != n) {
            outFile.write((char*) &obj,
sizeof(obj));
        }
    }

    inFile.close();
    outFile.close();
    remove("student.dat");
```

```
            case 6:
                return 0;
            }
        } while (ch != 6);
    }
```

THANK YOU!!

SS Barde
7769974262
ssbarde@pixaflip.com