Strassens Matrix Multiplication

$$\begin{array}{cc} A & B \\ [\quad] & \times [\quad] \\ m \times n & n \times r \\ 2\times 2 & 2 \times 2 \end{array}$$

$$C_{ij} = \sum_{k=1}^{n} A_{ik} * B_{kj}$$

<u>Normal Method</u>

```
for(i=0; i<n; i++)
{
  for(j=0; j<n; j++)
  {
    c[i,j] = 0;
    for(k=0; k<n; k++)
    {   c[i,j]+= A[i,k] * B[k,j];
    }
  }
}
```

$O(n^3)$

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \times B \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} = C \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix}$$

$C_{11} = a_{11} * b_{11} + a_{12} * b_{21}$ — (1)

$C_{12} = a_{11} * b_{12} + a_{12} * b_{22}$ — (2)

$C_{21} = a_{21} * b_{11} + a_{22} * b_{21}$ — (3)

$C_{22} = a_{21} * b_{12} + a_{22} * b_{22}$ — (4)

Scanned by CamScanner

$$A = \left[\begin{array}{cc|cc} a_{11} & a_{12} & a_{13} & a_{14} \\ \multicolumn{2}{c|}{A_{11}} & \multicolumn{2}{c}{A_{12}} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ \hline a_{31} & a_{32} & a_{33} & a_{34} \\ \multicolumn{2}{c|}{A_{21}} & \multicolumn{2}{c}{A_{22}} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{array}\right] \times B = \left[\begin{array}{cc|cc} b_{11} & b_{12} & b_{13} & b_{14} \\ \multicolumn{2}{c|}{B_{11}} & \multicolumn{2}{c}{B_{12}} \\ b_{21} & b_{22} & b_{23} & b_{24} \\ \hline b_{31} & b_{32} & b_{33} & b_{34} \\ \multicolumn{2}{c|}{B_{21}} & \multicolumn{2}{c}{B_{22}} \\ b_{41} & b_{42} & b_{43} & b_{44} \end{array}\right]$$

4×4 $\qquad$ 4×4

MM (A, B, n)

{

if (n ≤ 2)

{

$c_{11} = a_{11} * b_{11} + a_{12} * b_{21}$

$c_{12} = a_{11} * b_{12} + a_{12} * b_{22}$

③

④

}

But strassen has given his own formulas of multiplications. ~~that don't~~

else

{

mid = n/2

MM(A_{11}, B_{11}, n/2) + MM(A_{12}, B_{21}, n/2)

MM(A_{11}, B_{12}, n/2) + MM(A_{12}, B_{22}, n/2)

MM(A_{21}, B_{11}, n/2) + MM(A_{22}, B_{21}, n/2)

MM(A_{21}, B_{12}, n/2) + MM(A_{22}, B_{22}, n/2)

}

}

$$T(n) = \begin{cases} T(1) & n \le 2 \\ 8T(n/2) + n^2 & n > 2 \end{cases}$$

$$T(n) = 8T(n/2) + n^2$$

$a = 8$

$b = 2$

$f(n) = n^2$

$\log_b^a = \log_2^8 = 3$

$n^k = n^2$

$\theta(n^3)$

$\theta(n^3)$.

Strassen's Multiplication (Reduction of those 8 multiplication)

$P = (A_{11} + A_{22}) \cdot (B_{11} + B_{22})$

$Q = (A_{21} + A_{22}) \cdot B_{11}$

$R = (B_{12} - B_{22}) \cdot A_{11}$

$S = (B_{21} - B_{11}) \cdot A_{22}$

$T = (A_{11} + A_{12}) \cdot B_{22}$

$U = (A_{21} - A_{11}) \cdot (B_{11} + B_{12})$

$V = (A_{12} - A_{22}) \cdot (B_{21} + B_{22})$

Now its changed,

$$T(n) = \begin{cases} 1 & n \le 2 \\ 7T(n/2) + n^2 & n > 2 \end{cases}$$

$\log_2^7 = 2.81$ & $k = 2$

$O(n^{\log_2 7}) = O(n^{2.81})$

$C_{11} = P + S - T + V$

$C_{12} = R + T$

$C_{22} = Q + S$

$C_{21} = P + R - Q + U$

Scanned by CamScanner

Matrix Chain Multiplication (MCM)

A_1	A_2	A_3	A_4
5X4	4X6	6X2	2X7

Q. Which pair should be selected such that total cost of multiplication is minimum?

$$T(n) = \frac{2^n C_n}{n+1}$$

Ans.

M

	1	2	3	4
1	0	120	88	158
2		0	48	104
3			0	84
4				0

S

	1	2	3	4
1		1	1	3
2			2	3
3				3
4				

for l=1

m[1,1] m[2,2] m[3,3] m[4,4]

$A_1 = A_2 = A_3 = A_4 = 0$

for l=2

m[1,2] ∴ $A_1 \times A_2 = 120$ (Write 1 at S also)

5X4 4X6

m[2,3] ∴ $A_2 \times A_3 = 48$

4X6 6X2

m[3,4] ∴ $A_3 \times A_4 = 84$

6X2 2X7

mycompanion

Scanned by CamScanner

Formula $\min \{ m[i,k] + m[k+1,j] + d_{i-1} * d_k * d_j \}$

for $l=3$

$m[1,3]$ i.e. $m[1,2,3]$

$A_1 \times (A_2 \times A_3)$ or $(A_1 \cdot A_2) A_3$

5×4 | 4×6 | 6×2

$m[1,1] + m[2,3] + \underbrace{5 \times 4 \times 2}_{\text{Final dimensions}}$

$m[1,2] + m[3,3] + \underbrace{5 \times 6 \times 2}_{\text{Final dimension}}$

$0 + 48 + 40$ — $120 + 0 + 60$

$88 < 180$

so take 88

$m[2,4]$

$A_2 \cdot (A_3 \cdot A_4)$ or $(A_2 \cdot A_3) \cdot A_4$

4×6 | 6×2 | 2×7 — 4×6 | 6×2 | 2×7

$m[2,2] + m[3,4] + 4 \times 6 \times 7$ — $m[2,3] + m[4,4] + 4 \times 2 \times 7$

$0 + 84 + 168$ — $48 + 0 + 56$

$252 > 104$

take this

for $l=4$

$m[1,4] = \min \begin{cases} ① \; m[1,1] + m[2,4] + 5 \times 4 \times 7 \\ ② \; m[1,2] + m[3,4] + 5 \times 6 \times 7 \\ ③ \; m[1,3] + m[4,4] + 5 \times 2 \times 7 \end{cases}$ All possibil

$\Rightarrow ① = 0 + 104 + 40$ — ③ $88 + 0 + 70 = 158$

② $= 120 + 84 + 210$ — so ③ is smallest

myCOMPANION

Longest Common Sequence

String1 : abcdefghij

String2 : cdgi (LCS)

String 1 : abcdefghij

String 2 : ecdgi so cfd skipped

⇓

egi, but cdgi is also holding

String 1 : abdace

String2 : babce only bace.

abdace

babce also abce

```
int LCS (i,j)
{
  if (A[i]=='\0' || B[j]=='\0')
      return 0;
  else if (A[i]==B[j])
      return 1+LCS(i+1,j+1);
  else
      return max(LCS(i+1,j), LCS(i,j+1));
}
```

mycompanion

Scanned by CamScanner

A

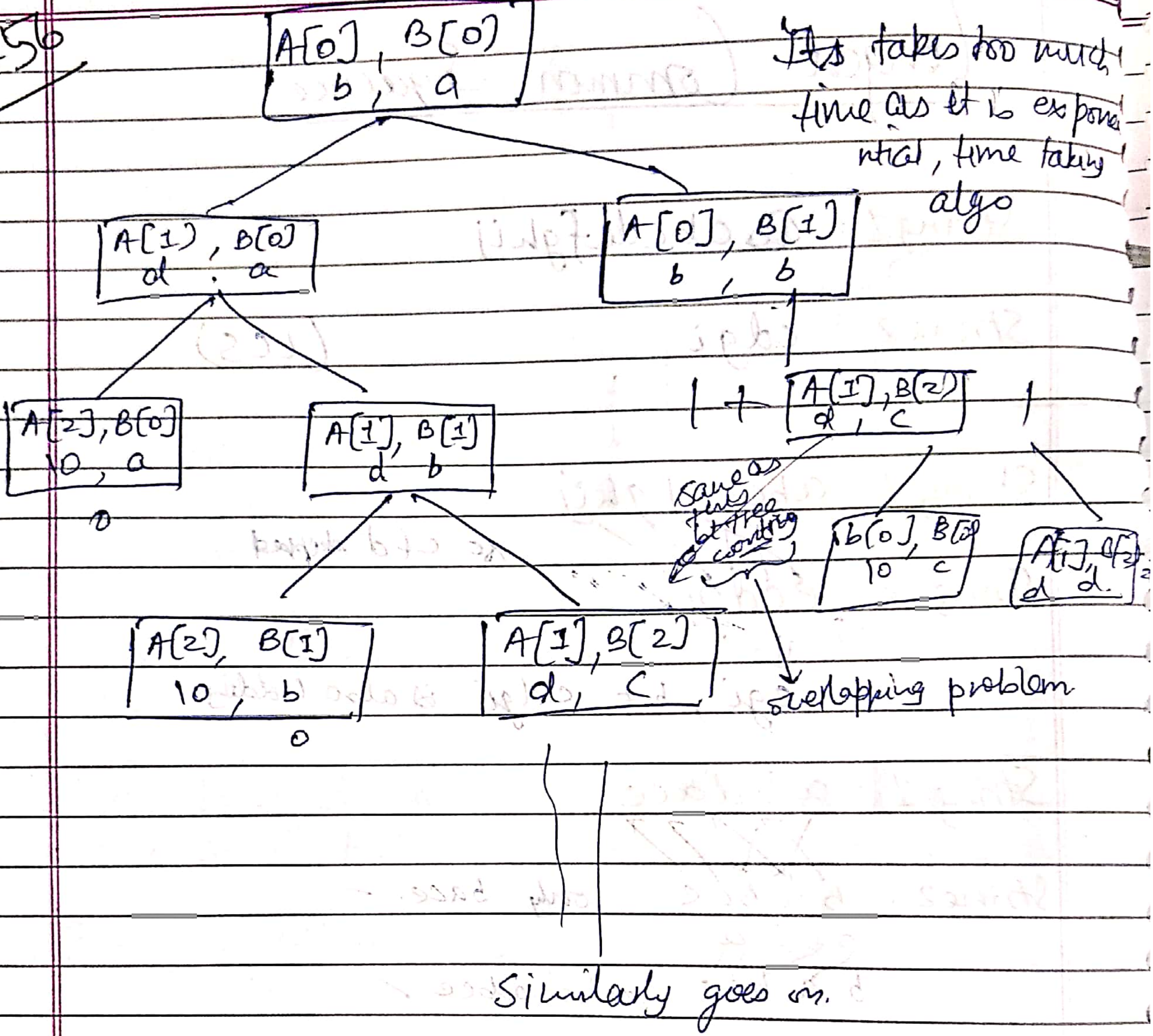
b	d	10
0	1	2

B

a	b	c	d	10
0	1	2	3	4

Video-56

Its takes too much time as it is exponential, time taking algo



So, if we store the steps of recursion, its called memorization and reduce time.

A↓ B→	a 0	b 1	c 2	d 3	10 4
b 0	2	2			
d 1	1	1	1	1	
10 2	0	0	0		0

fill these values using the above tree and no need to make function calls done at the right side of tree.

Time ⇒ (m×n) in O(m×n)

mycompanion

Scanned by CamScanner

Eg.

A | b | d |, indices 1 2

B | a | b | c | d |, indices 1 2 3 4

Using DP

if $(A[i] = B[j])$

$$LCS[i,j] = 1 + LCS[i-1, j-1]$$

else

$$LCS[i,j] = \max(LCS[i-1,j],\ LCS[i,j-1])$$

$O(m \times n)$

		ϕ	a	b	c	d
A ↓ / B →		0	1	2	3	4
	0	0	0	0	0	0
b	1	0	0	1	1	1
d	2	0	0	1	1	2

b d

so $\boxed{bd}$

Eg

A = | s | t | o | n | e |, indices 1 2 3 4 5

B = | l | o | n | g | e | s | t |, indices 1 2 3 4 5 6 7

			l	o	n	g	e	s	t
A ↓ / B →		0	1	2	3	4	5	6	7
	0	0	0	0	0	0	0	0	0
s	1	0	0	0	0	0	0	1	1
t	2	0	0	0	0	0	0	1	2
o	3	0	0	1	1	1	1	1	2
n	4	0	0	1	2	2	2	2	2
e	5	0	0	1	2	2	3	3	3

o n e

(one) LCS

myCompanion

Scanned by CamScanner

Eg

abcdaf
acbcf

		a	b	c	d	a	f
	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
a 1	0	1	1	1	1	1	1
c 2	0	1	1	2	2	2	2
b 3	0	1	2	2	2	2	2
c 4	0	1	2	3	3	3	3
f 5	0	1	2	3	3	3	4

a b c f

Longest Common Sequence (abcf)

0-1 Knapsack Problem

$x_i \{0/1\}$

$P = \{1, 2, 5, 6\}$ Profit
$w = \{2, 3, 4, 5\}$ weight

$n = 4$
$m = 8$

$\sum p_i x_i \rightarrow$ Max.

$\sum w_i x_i \leq m$ → capacity of bag (weight)

Video - 51

→ It will solved in a sequence of decisions and decisions are taken from last object to first.

→ Try all possible solⁿ and pick up the best one.

$x = \{0, 0, 0, 1\}$ only last object is taken

$\{ 0 \rightarrow$ Not taken
$1 \rightarrow$ taken

hence, $O(2^n)$ exponential (Time consuming)

Tabulation Method (DP)

(Capacity of bag) $m = 8$

m (weight)

Profit	Weight	V	0	1	2	3	4	5	6	7	8
		0	0	0	0	0	0	0	0	0	0
1	2	1	0	0	1	1 ③	1	1	1	1	1
2	3	2	0	0	1	2 ①	2	3	3	3	3
5	4	3	0	0	1	2 ②	5	5	6	7	7
6	5	4	0	0	1	2	5	6	6	7	8

Max profit

x_1	x_2	x_3	x_4
0	1	0	1

Formula

$$V[i,w] = \max\{V[i-1,w],\ V[i-1, w-w[i]] + P[i]\}$$

(row 4)

Now, to fill ($x_1 x_2 x_3 x_4$), we check last row, whose max profit is, here ⇒ 8. so include it hence $x_4 = 1$. Profit $(8-6) = 2$ so remaining profit ⇒ 2. Now check row 3, 2② matches the profit but just above it we also have 2①. so, we dont consider it, and move to row 2. Now, again check for 2①, we find ① but just above 1 i.e. ③, we have 1 that means there is profit from it. so consider row 2. similarly go on

Eg

P = | 0 | 1 | 2 | 5 | 6 |

0 1 2 3 4

W = | 0 | 2 | 3 | 4 | 5 |

$n=4$

$m=8$

P	W	V \ W / i	0	1	2	3	4	5	6	7	8
		0	0	0	0	0	0	0	0	0	0
1	2	1	0	0	1	1	1	1	1	1	1
2	3	2	0	0	1	2	2	3	3	3	3
5	4	3	0	0	1	2	5	5	6	7	7
6	5	4	0	0	1	2	5	6	6	7	8

x_1	x_2	x_3	x_4
0	1	0	1

so $8-6=2$

$2-2=0$

Eg-

$m=7$ P = {1, 4, 5, 7}

$n=4$ W = {1, 3, 4, 5}

P	W		0	1	2	3	4	5	6	7
		0	0	0	0	0	0	0	0	0
1	1	1	0	1	1	1	1	1	1	1
4	3	2	0	1	1	4	5	5	5	5
5	4	3	0	1	1	4	5	6	6	9
7	5	4	0	1	1	4	5	7	8	9

$9-7 \Rightarrow 4$

$4-4 \Rightarrow 0$

x_1	x_2	x_3	x_4
0	1	0	

Scanned by CamScanner

Algo

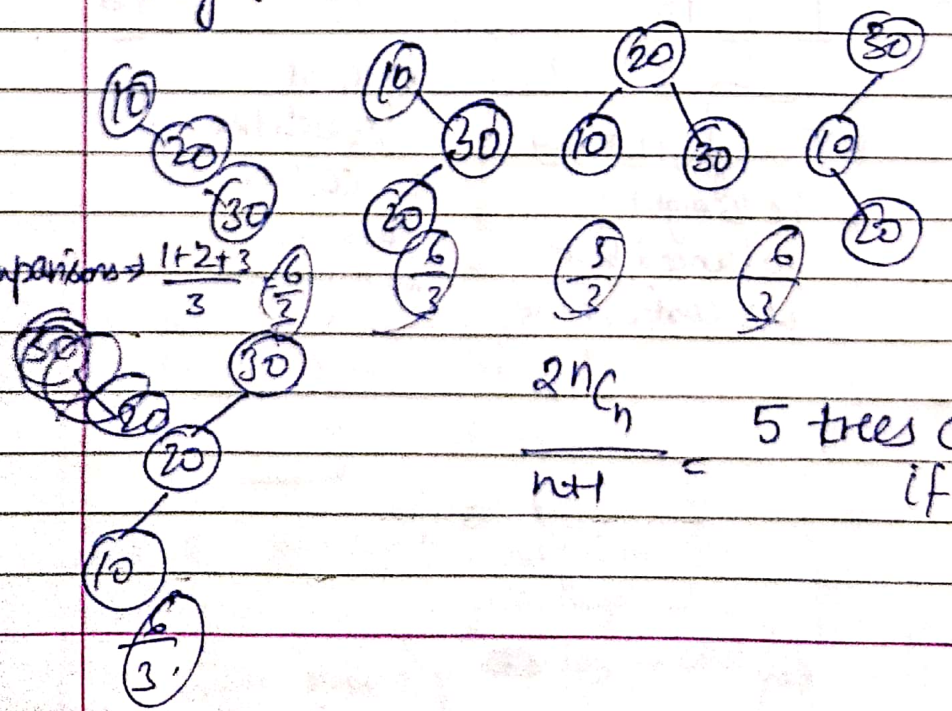
```
{
  if (i==0 || w==0)
      K[i][w]=0;
  else if (wt[i] <= w)
  {
   k[i][w] = max[P[i] + k[i-1][w-w[i]], k[i-1][w]
  }
  else
  {
   k[i][w] = k[i-1][w];
  }
}
```

OPTIMAL BINARY SEARCH TREE

Keys(n) = 10, 20, 30

logn = height of tree

O(logn)
Time taken to search in BST



Comparisons → $\frac{1+2+3}{3}$

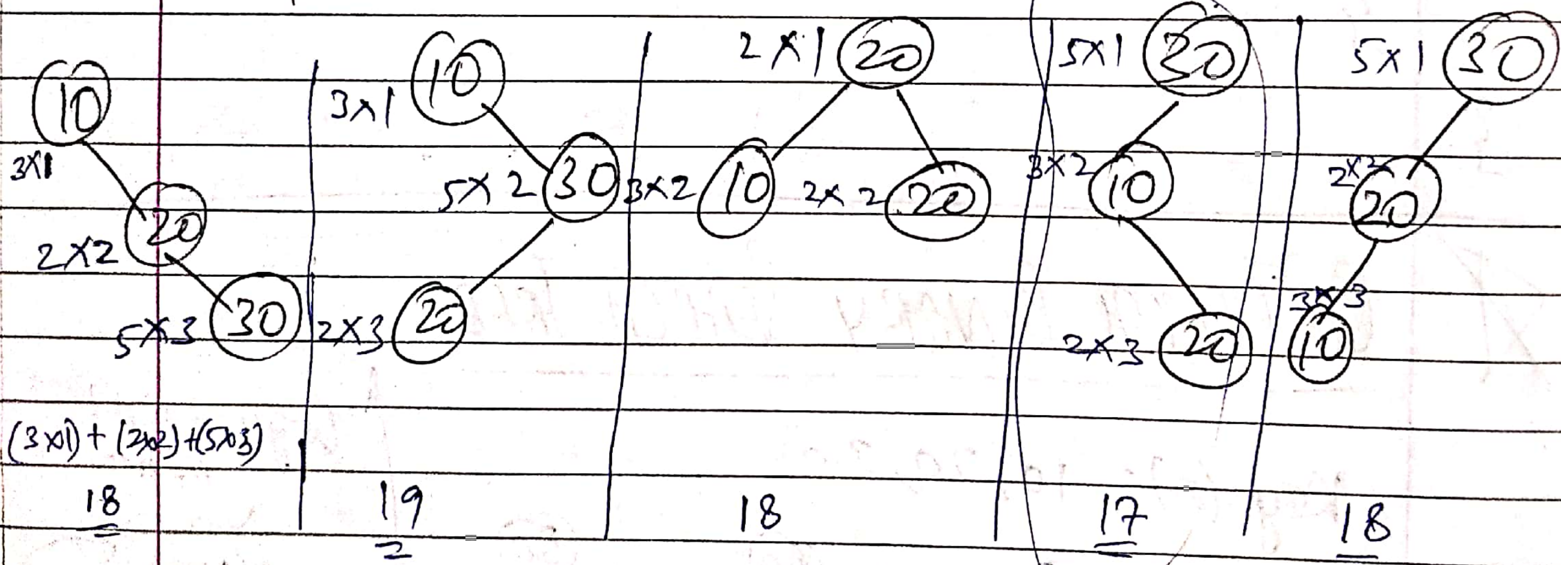
$\frac{{}^{2n}C_n}{n+1}$ = 5 trees can be made if n=3

So as height of 3rd tree is less, so it does less comparisons and its called balanced binary search tree.

Now, if we want to search ~~trees~~ keys in trees, but their frequency of search is also given and we search those with more frequency more often. Hence, we find which tree organisation will be optimal to find the key.

So, keys = 10, 20, 30
frequency ⇒ 3, 2, 5

No. of trees ⇒



Though, it may be height balanced but its cost is more

Optimal search tree as, cost of this tree is minimum.

Eg

$l/n =$	1	2	3	4
Keys ⇒	10	12	16	21
r ⇒	4	2	6	3

for $l=1$

	1	2	3	4
1	4			
2		2		
3			6	
4				3

Now, for = $l=2$

10	12	16	21
4	2	6	3

for $l=2$,

for $l=2$

	1	2	3	4
1	4			
2		2		
3			6	
4				3

C[1, 2] – 1
C[2, 3] – 2
C[3, 4] – 3

1

$n=$	1	2	3	4
	10	12	16	21
	4	2	6	3

Adding r + min { tree cost made from each key. }

6 + min { key → cost } ⇒

⇒ 6 + min { 1 → 2, 2 → 4 } cost

⇒ As 6 + 2 = 8 ✓

but 6 + 4 = 10 so we take 8.

Scanned by CamScanner

key indication

	1	2	3	4
1	4	8^1		
2		2	10^3	
3			6	12^3
4				3

② $C[2,3]$

1	2	3	4
10	12	16	21
4	2	6	3

Adding ? + min { tree cost made from each key }

$$8 + \min\left\{ \begin{array}{l} 2 \rightarrow 6 \\ 3 \rightarrow 2 \end{array} \right\}$$

$8 + 6 = 14$

but $8 + 2 = 10$ ✓

③ $C[3,4]$

1	2	3	4
10	12	16	21
4	2	6	3

Adding 9 + { key → cost }

$$9 + \min\left\{ \begin{array}{l} 3 \rightarrow 3 \\ 4 \rightarrow 6 \end{array} \right\}$$

$9 + 3 = 12$ ✓

$9 + 6 = 15$

Scanned by CamScanner

for $l = 3$

$C[1,3] \rightarrow$ 4

$C[2,4] \rightarrow$ 5

	1	2	3	4
1	4	8^1	20^3	
2		2	10^3	16^3
3			6	12^2
4				3

4 $C[1,3]$

1	2	3	4
10	12	16	21
4	2	6	3

$$12 + \min \begin{cases} 1 \rightarrow C[2,3] \\ 2 \rightarrow C[1,1] + C[3,3] \\ 3 \rightarrow C[1,2] \end{cases}$$

$$12 + \min \begin{cases} 1 \rightarrow 10 \\ 2 \rightarrow 10 \\ 3 \rightarrow 8 \end{cases}$$

$12 + 10 = 22$

$12 + 10 = 22$

hence $\Rightarrow 12 + 8 \Rightarrow 20$

5 $C[2,4]$

1	2	3	4
10	12	16	21
4	2	6	3

$$11 + \min \begin{cases} 2 \rightarrow 12 \\ 3 \rightarrow 5 \\ 4 \rightarrow 10 \end{cases}$$

$11 + 12$ ✗

$11 + 10$ ✗

so $11 + 5 \Rightarrow 16$

Scanned by CamScanner

for l=4

(6)

	1	2	3	4
	10	12	16	21
	4	2	6	3

	1	2	3	4
1	4	8^1	20^3	26 (3)
2		2	10^3	16^3
3			6	12^3
4				3

C[1,4] – (6)

(6)

$$15 + \begin{cases} 1 \to C[2,4] \\ 2 \to C[1,1] + C[3,4] \\ 3 \to C[1,2] + C[4,4] \\ 4 \to C[1,3] \end{cases}$$

$$15 + \begin{cases} 1 \to 16 \\ 2 \to 4+12 \\ 3 \to 8+3 \\ 4 \to 20 \end{cases} \Rightarrow 15 + \min \begin{cases} 1 \to 16 \\ 2 \to 16 \\ 3 \to 11 \\ 4 \to 20 \end{cases}$$

$15 + 11 = 26$

Scanned by CamScanner