Lecture-3(a)

Coin Change:

1. Using DP:

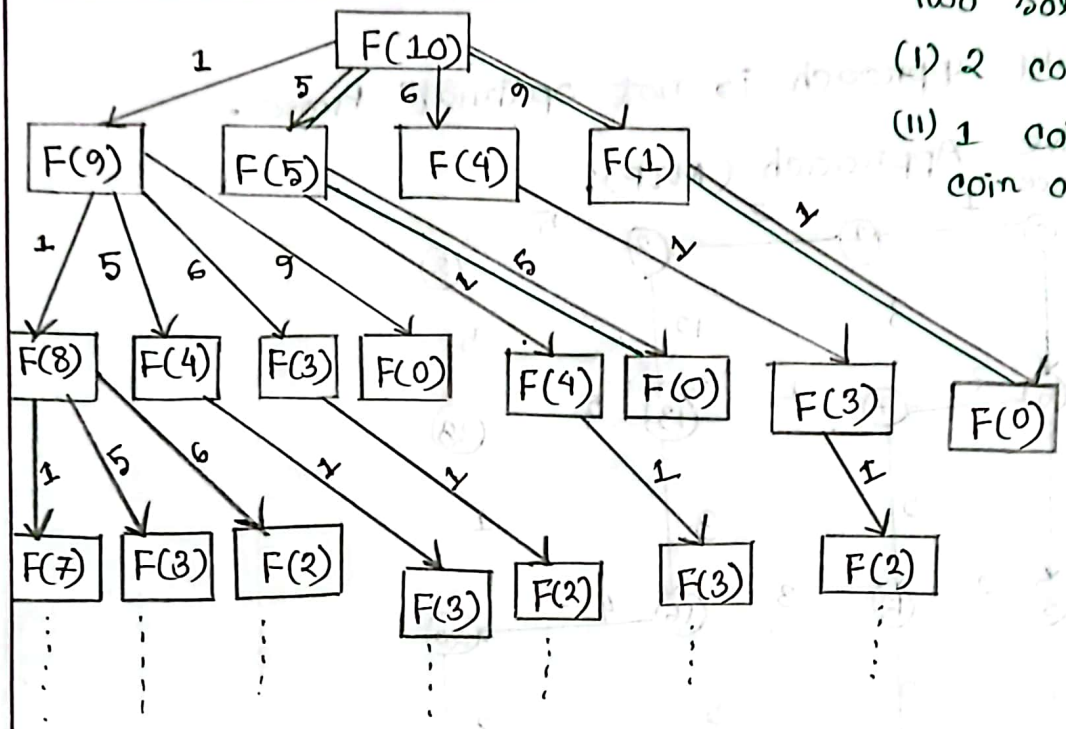
$C = \{1, 5, 6, 9\}$ Sum = 10

	0	1	2	3	4	5	6	7	8	9	10
1	0	1	2	3	4	5	6	7	8	9	10
5	0	1	2	3	4	min(5, 1+0) 1	min(6, 1+1) 2	min(7, 1+2) 3	min(8, 1+3) 4	min(9, 1+4) 5	min(10, 1+4) 2
6	0	1	2	3	4	1	min(2, 1+0) 1	min(3, 1+1) 2	min(4, 1+2) 3	min(5, 1+3) 4	min(2, 1+4) 2
9	0	1	2	3	4	1	1	2	3	min(4, 1+0) 1	min(2, 1+1) 2

$5 - 5 = 0$

Solution : 2 coin of "5"

(II) Recursive Solution:



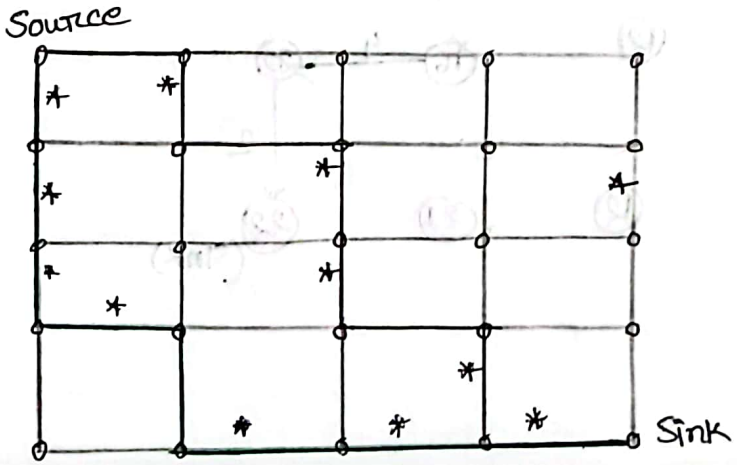
Two solve:
(I) 2 coin of "5"
(II) 1 coin of "9" & 1 coin of "1"

Manhattan Tourist Problem (MTP):

Seeking a path (from source to sink/destination) to travel (only eastward and southward) with most number of attractions (*) in the Manhattan grid.

Constraint: Only eastward & southward.

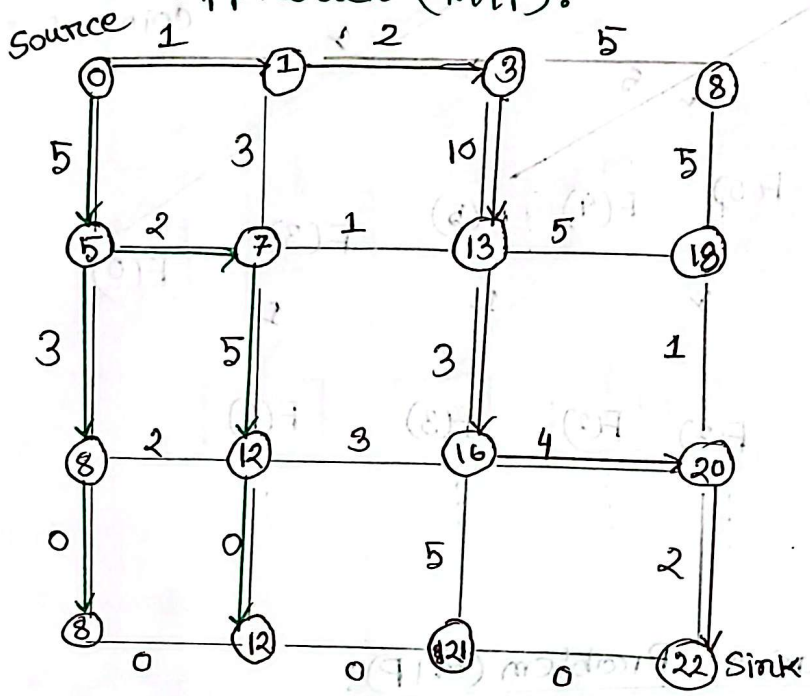
- The graph will always be directed acyclic graph.



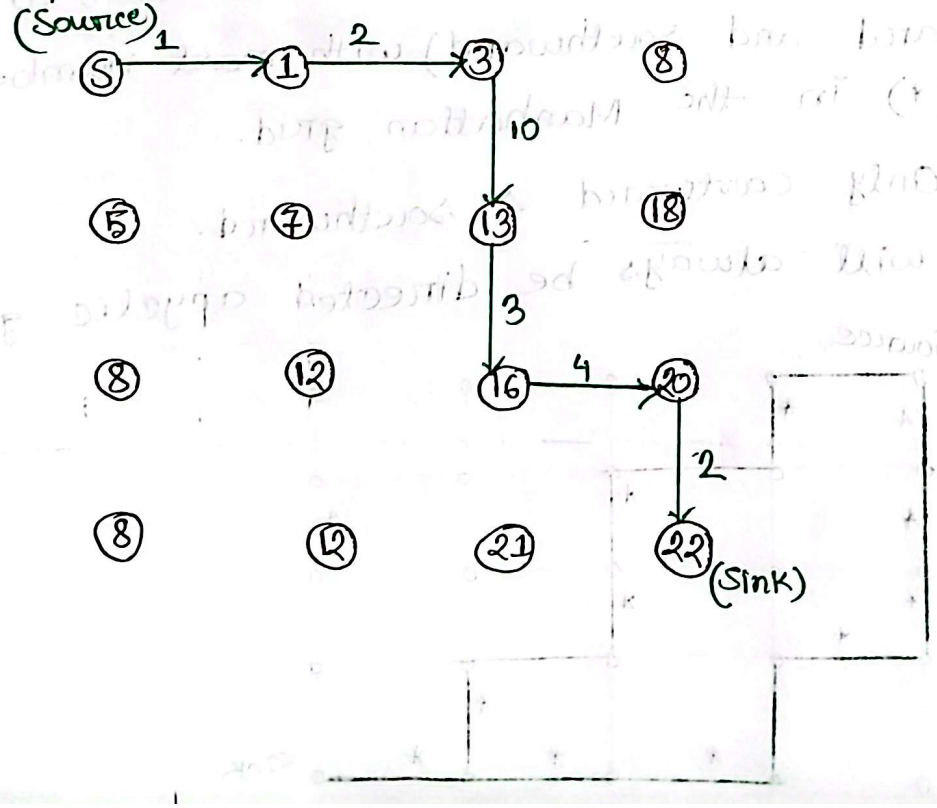
- Every intersection is a node and path is edge.

"Greedy approach is not optimal" Prove.

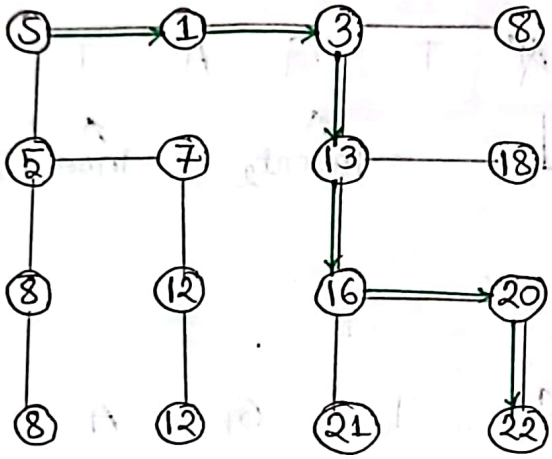
Recursive Approach (MTP):



Graph:



Dynamic Programming (MTP):



Alignment : 2 row representation:

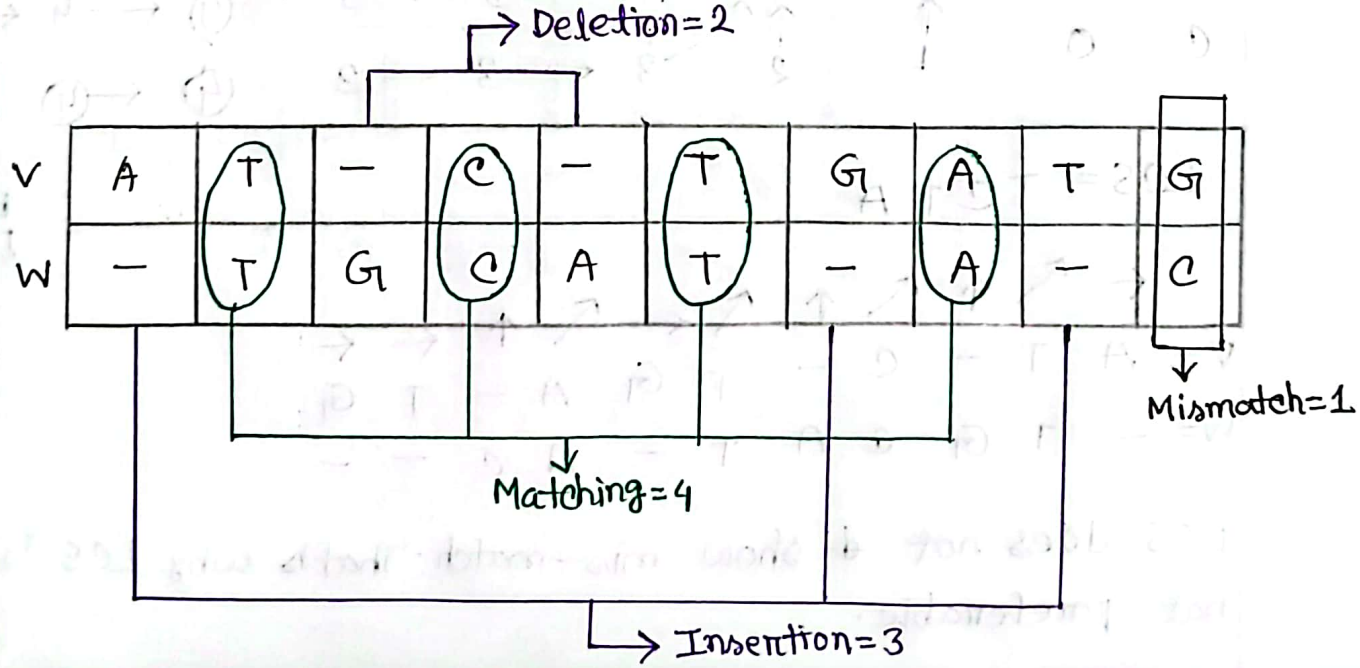
V = A T C T G A T G n=8

W = T G C A T A C m=7

Let,

upper gap = deletion

lower gap = insertion



V=	A	T		C		T	G	A	T	G
W=	A	T	~~G~~	C	~~A~~	T	G	A	T	C

Insert$_1$ · Delete=2 · Insert$_2$ · Insert$_3$

V

W \ V	–	A	T	C	T	G	A	T	G
–	0	0	0	0	0	0	0	0	0
T	0	0	1	1	1	1	1	1	1
G	0	0	1	1	1	2	2	2	2
C	0	0	1	2	2	2	2	2	2
A	0	1	1	2	2	2	3	3	3
T	0	1	2	2	3	3	3	4	4
A	0	1	2	2	3	3	4	4	4
C	0	1	2	3	3	3	4	4	4

W

Edit Distance

LCS = T C T A

V = A T – C – T G A – T G

W = – T G C A T – A C – –

LCS does not show mis-match. That's why LCS is not preferable.

V = A T C T G A T C

W = T G C A T A C

V

W \ V	–	A	T	C	T	G	A	T	C
–	(0)	0	0	0	0	0	0	0	0
T	0	0	(1)	1	1	1	1	1	1
G	0	0	1	1	1	2	2	2	2
C	0	0	1	(2)	2	2	2	2	3
A	0	1	1	(2)	2	2	3	3	3
T	0	1	2	2	(3)	3	3	4	4
A	0	1	2	2	3	3	(4)	(4)	4
C	0	1	2	3	3	3	4	4	(5)

W

Edit Distance

LCS = T C T A C

← ↖ ↑ ↖ ↑ ↖ ← ↑ ← ↖

V = A T – C – T G A T C

W = – T G C A T – A – C

Hamming Distance (Aligning Sequence without insertion and deletion):

P = N A F I S A

Q = N A B I L A

$d_H = (0+0+1+0+1+0) = 2$

~~Sequence~~

- If two sequence's character match than '0' otherwise '1'.

Edit Distance:

- Index by index always তুলনা করবেনা।
- Count the insertion & deletion.

i^{th} letter of V with

J^{th} letter of W

V = A T A T A T A T

W = T A T A T A T A

$d_H = (1+1+1+1+1+1+1+1)$

$= 8$

If shift W by 1

V = A T A T A T A T —

W = — T A T A T A T A

$d_H = (1+0+0+0+0+0+0+0+1)$

$= 2$

Formula of Edit distance:

Edit distance $(v, w) = |v| + |w| - 2|LCS(v, w)|$

↳ distance of v & w will be count without gap.

TGCATAT → ATCCGAT in 5 steps

1. TGCATA T → delete last T
2. TGCATA → delete last A
3. TGCAT → insert A at front
4. ATGCAT → substitute C for 3rd G
5. ATCCAT → insert G before last A
6. ATCCGAT → Done

Lecture-3(b)

Indel → Insertion or deletion

Global alignment:

$$S_{i,j} = \max \begin{cases} S_{i-1,j-1} + 1 & \text{if } v_i = w_i \\ S_{i-1,j-1} - \mu & \text{if } v_i \neq w_i \\ S_{i-1,j} - \sigma \\ S_{i,j-1} - \sigma \end{cases} \} \text{ Indel}$$

Scoring Schema:

+1 : match premium

$-\mu$: mismatch penalty

$-\sigma$: indel penalty

#We scoring 1 for match & zero (0) for indel.

#match $-\mu$ (#mismatch) $-\sigma$ (# indel)

Scoring Matrix (Mismatch count):

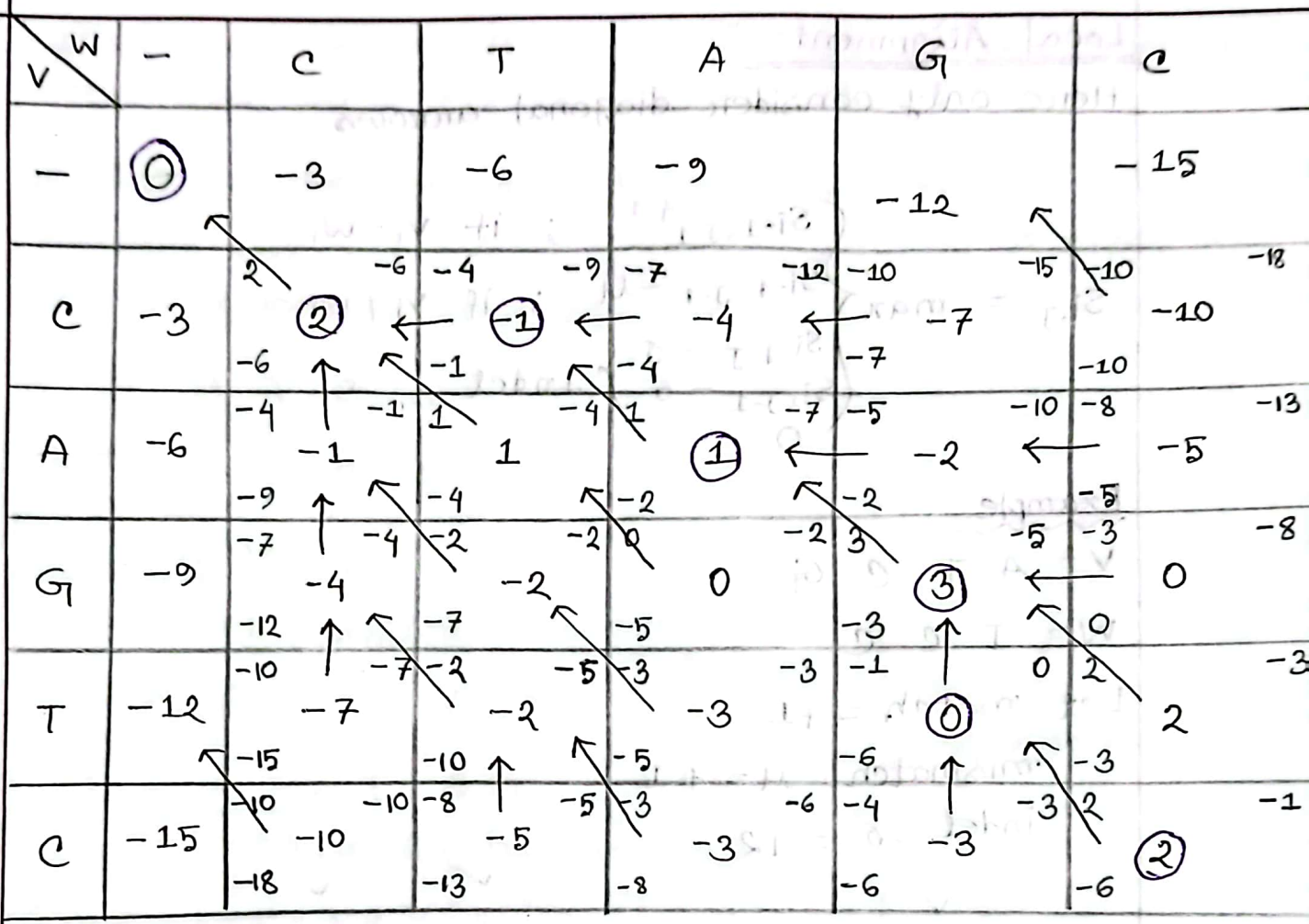
W = C T A G C

V = C A G T C

Let, match = +2

mismatch = $-1 = -\mu$

Indel = $-3 = -\sigma$



↖ ← ↖ ↑ ↑ ↖

W = C T A G — C

V = C — A G T C

Score = 2 − 3 + 2 + 2 − 3 + 2

= 2 = Final Score.

Local Alignment:

Here only consider diagonal arrows.

$$S_{i,J} = \max \begin{cases} S_{i-1,J-1} + 1 & ; \text{ if } v_i = w_i \\ S_{i-1,J-1} - \mu & ; \text{ if } v_i \neq w_i \\ S_{i-1,J} - \sigma \\ S_{i,J-1} - \sigma \end{cases} \text{Indel}$$
$$0$$

Example:

V = A T C G

W = T C C

Let, match = +1

mismatch, $\mu = +1$

indel, $\sigma = +2$

	W \ V	–	A	T ✓	C ✓	G
	–	0	0	0	0	0
✓	T	0	-1 -2 / -2 : 0	1 -2 / -2 : ①	-1 -2 / -1 : 0	-1 -2 / -2 : 0
✓	C	0	-1 -2 / -2 : 0	-1 -1 / -2 : 0	2 -2 / -2 : ②	-1 -2 / 0 : 0
	C	0	-1 -2 / -2 : 0	-1 -2 / -2 : 0	1 0 / -2 : 1	1 -2 / -1 : 1

- এখানে কোনো box এ zero বা (-) value হলে আমরা zero বসাবো। এবং নাহলে max value হবে।

Find out the max and diagonal from it.

Multiple max value হলে সবচেয়ে নিচের row তে right এর column এ max choose করতে হবে।

```
V =  A  T  C  G
         |  |
W =  T  C  C
```

Gap Penalties:

1. Constant: (consider consecutive gap as 1 gap)

```
T  A  C  C  T  A  G
T  _  _  _  T  A  _
   └──┘
   3 consecutive gaps as 1.
```

Score = -1 + 1 + 1 - 1

= 1

Let,
match = +1
gap/indel = -1

2. Linear (Least Realistic):

```
T  A  C  C  T  A  G
T  _  _  _  T  A  _
```

(Consider every gap individually).

Score = -1 - 1 - 1 + 1 + 1 - 1

= -1

Let,
match = +1
gap/indel = -1

3. Affine:

- Gap opening: multiple gap পরপর হলে 1st gap গুলো,
- Gap Extention: multiple gap এর Gap opening বাদে বাকি গুলো।

T A C C T A G

T

gap extention penalty (σ)

Gap opening penalty (ρ)

Let,
match = +1, $\rho = -2$, $\sigma = -1$

Score = $1 - 2 - 1 - 1 + 1 + 1 - 2$

$= -3$

Gap penalty = $\rho + \sigma(k-1)$

$\Rightarrow k$ = number of consecutive gap

if $k=3$, gap penalty = $-2 + (-1)(3-1)$

$= -2 + (-1)(2)$

$= -2 - 2 = -4$

if $k=1$, gap penalty = $-2 + (-1)(1-1)$

$= -2 + (-1)(0)$

$= +2 + 0$

$= -2$