11/01/2023 Wednesday

Lecture No. 22

→ DYNAMIC PROGRAMMING:

* Word Split / Break Problem:

Given a string and English vocabulary can this string be splitted into multiple words in such a way that each word belong to the vocabulary.

Exp:

I|am|king True.

I|a m|k False.

1. $\overset{0}{I}\,\overset{1}{a}\overset{2}{m}\,\overset{3}{k}\overset{4}{i}\overset{5}{n}\overset{6}{g}$

```
if (str [i, j] is in vocabulary)
    MAT[i, j] = T.
else
    MAT[i, j] = T
    if there exists a k such that
  MAT[i,k] && MAT[k+1, j]
```

0	J	T
1	a	T
2	m	F
3	k	F
4	i	T
5	n	F
6	g	F

0123456
Jamking

	0	1	2	3	4	5	6
0	T	T	T^{0}	F	F	T^{0}	T^{0}
1		T	T	F	F	T^{2}	T^{2}
2			F	F	F	F	F
3				F	F	~~F~~$T^{(3-5)}$	$T^{(3-6)}$
4					T	T	F
5						F	F
6							F

MAT[0,3] = MAT[0,0] && MAT[1,3] → F
= MAT[0,1] && MAT[2,3] → F
= MAT[0,2] && MAT[3,3] → F

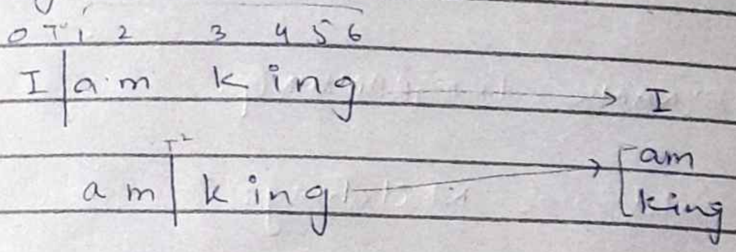
k = 1 MAT[1,4] = Mat[1,1] && MAT[2,4] = T && T = F
k = 2 MAT[1,4] = MAT[1,2] && MAT[3,4] = T && F = F
k = 3 MAT[1,4] = MAT[1,3] && MAT[4,4] = F && F = F

k = 1 MAT[1,5] = MAT[1,1] && MAT[2,5] = T && F = F
k = 2 MAT[1,5] = MAT[1,2] && MAT[3,5] = T && T = T.
k = 3 MAT[1,5] = MAT[1,3]
k = 4 MAT[1,5] = MAT[1,4]
No Need To compute.

k = 0 MAT[0,5] = Mat[0,0] && MAT[1,5] = T && T = T^{0}

k = 1 MAT[1,6] = MAT[1,1] && MAT[2,6] = T && F = F
k = 2 MAT[1,6] = MAT[1,2] && MAT[3,6] = T && T = T^{2}

→ BackTracking



Exp#2.

0 1 2 3 4 5
u r king

	0	1	2	3	4	5
0	F	F	F	F	F	F
1		F	F	F	F	F
2			F	F	$T^{(2-4)}$	$T^{(2-5)}$
3				T	$T^{(3-4)}$	F
4					F	F
5						F

MAT[0,2] = MAT[0,0] && MAT[1,2] = F&&F = f.   k=0
MAT[0,2] = MAT[0,1] && MAT[2,2] = F&&F = F   k=1

MAT[0,5] = MAT[0,0] && MAT[1,5] = F&&F = F   k=0
" = MAT[0,1] && MAT[2,5] = F&&T = F   =1
" = MAT[0,2] && MAT[3,5] = F&&F = F   =2
" = MAT[0,3] && MAT[4,5] = F&&F = F   =3
" = MAT[0,4] && MAT[5,5] = F&&F = F   =4

MAT[0,4] = MAT[0,0] && MAT[1,4] = F&&F = F   k=0
= MAT[0,1] && MAT[2,4] = F&&T = F   k=1
= MAT[0,2] && MAT[3,4] = F&&T = F   k=2
= MAT[0,3] && MAT[4,4] = F&&F = F   k=3

12/01/2023. Friday

Lec No. 23

→ DYNAMIC PROGRAMMING:

- Wild Card Matching:

→ Decision Problem (Yes OR NO).

→ * = 0 or more characters

? = any one character

Pattern	Strings
a * b	a, x, y, b ab, aab, abab,.... = T.
↓ Starting char 'a' ending char 'b' in middle there can be any 0 or more character.	b, a, ac, abc ..... = F ↳ Because doesn't match the pattern.
a ? b ↳ any one char.	aab, abb, axb .... = T. ab, ac, b, acd, a, acdb, ... = F.
Exp: x ? y * z	xayz, xxyyzz, xaybcz ... = T xaz, ayz, xyz, xay, xbc ... = F

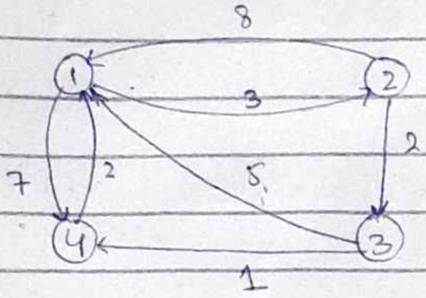
Pattern : x ? y * z
string : x a y l m z

	0	1	2	3	4	5
		x	?	y	*	z
0	T	F	F	F	F	F
1 x	F	T	F	F	F	F
2 a	F	F	T	F	F	F
3 y	F	F	F	T	T	F
4 l	F	F	F	F	T	F
5 m	F	F	F	F	T	F
6 z	F	F	F	F	T	Ⓣ

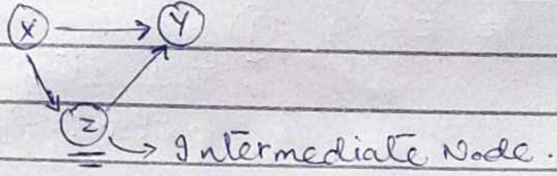
The string matches the pattern.

```
if(str[i] == pattern[j] || pattern[j]='?')

MAT[i,j] = MAT[i-1,j-1]

else if (pattern[j]=='*')
    MAT[i,j]= MAT[i-1,j] || Mat[i,j-1]
else
    MAT[i,j] = 'F'
```

- Floyd Warshall's Algorithm:
  ↳ All pair shortest Path.



- Intermediate Node



$A^0 \rightarrow$ Original matrix.

$$A^0 = \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 3 & \infty & 7 \\ 2 & 8 & 0 & 2 & \infty \\ 3 & 5 & \infty & 0 & 1 \\ 4 & 2 & \infty & \infty & 0 \end{array}$$

via Intermediate Node 1.

$$A^1 = \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 3 & \infty & 7 \\ 2 & 8 & 0 & 2 & 15 \\ 3 & 5 & 8 & 0 & 1 \\ 4 & 2 & 5 & \infty & 0 \end{array}$$

$$A^1[2,3] = \min \begin{cases} A^0[2,3] = 2 \\ A^0[2,1] + A^0[1,3] = 8 + \infty \end{cases}$$

$$= \boxed{2}$$

$$A^1[2,4] = \min \begin{cases} A^0[2,4] = \infty \\ A^0[2,1] + A^0[1,4] = 8 + 7 = 15 \end{cases}$$

$$= \boxed{15}$$

$$A^1[3,2] = \min \begin{cases} A^0[3,2] = \infty \\ A^0[3,1] + A^0[1,2] = 5 + 3 = 8 \end{cases}$$

$$= \boxed{8}$$

$$A^1[3,4] = \min \begin{cases} A^0[3,4] = 1 \\ A^0[3,1] + A^0[1,4] = 5 + 7 = 12. \end{cases}$$

$$= \boxed{1}$$

$$A^1[4,2] = \min \begin{cases} A^0[4,2] = \infty \\ A^0[4,1] + A^0[\cancel{2}1,2] = 2 + 3 \end{cases}$$

$$= \boxed{5}$$

$$A^1[4,3] = \min \begin{cases} A^0[4,3] = \infty \\ A^0[4,1] + A^0[1,3] = 2 + \infty \end{cases}$$

$$= \infty$$

$$A^2 = \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 3 & 5 & 7 \\ 2 & 8 & 0 & 2 & 15 \\ 3 & 5 & 8 & 0 & 1 \\ 4 & 2 & 5 & 7 & 0 \end{array}$$

$$A^2[1,3] = \min \begin{cases} A^1[1,3] = \infty \\ A^1[1,2] + A^1[2,3] = 3+2 = \textcircled{5} \end{cases}$$

$$A^2[1,4] = \min \begin{cases} A^1[1,4] = \textcircled{7} \\ A^1[1,2] + A^1[2,4] = 3+15 = 18 \end{cases}$$

$$A^2[3,1] = \min \begin{cases} A^1[3,1] = \textcircled{5} \\ A^1[3,2] + A^1[2,1] = 8+8 = 16 \end{cases}$$

$$A^2[4,1] = \min \begin{cases} A^1[4,1] = \textcircled{2} \\ A^1[4,2] + A^1[2,1] = 5+8 = 13. \end{cases}$$

$$A^3 = \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 3 & 5 & 6 \\ 2 & 7 & 0 & 2 & 3 \\ 3 & 5 & 8 & 0 & 1 \\ 4 & 2 & 5 & 7 & 0 \end{array}$$

$$A^3[1,2] = \min \begin{cases} A^2[1,2] = \textcircled{3} \\ A^2[1,3] + A[3,2] = 5+8 = 13 \end{cases}$$

$$A^3[1,4] = \min \begin{cases} A^2[1,4] = 7 \\ A^2[1,3] + A^2[3,4] = 5 + 1 = \textcircled{6}. \end{cases}$$

$$A^4 = \begin{array}{c|cccc|} 1 & 0 & 3 & 5 & 6 \\ 2 & 5 & 0 & 2 & 3 \\ 3 & 3 & 6 & 0 & 1 \\ 4 & 2 & 5 & 7 & 0 \end{array} \rightarrow$$ Final Sol.

Shortest path b/w every vertices

$$A^4[3,2] = \min \begin{cases} A^3[3,2] = 8 \\ A^3[3,4] + A^3[4,2] = 1 + 5 = \textcircled{6}. \end{cases}$$

- Formula:

$$A^k[i,j] = \min \begin{cases} A^{k-1}[i,j] \\ A^{k-1}[i,k] + A^{k-1}[k,j]. \end{cases}$$

- Time complexity:

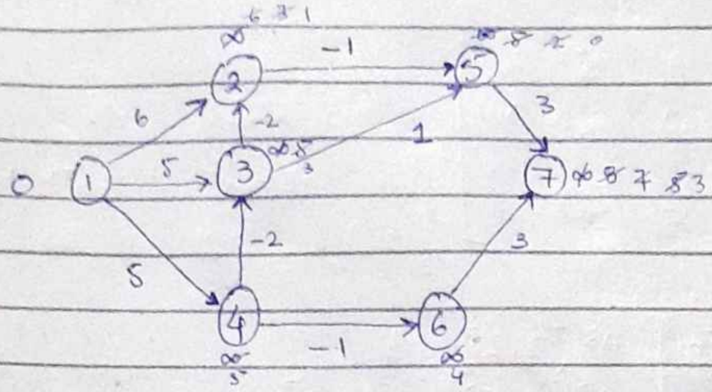
$$= \boxed{O(n^3)}$$

Three loops outer loop k

2 loops for matrices i, j

* * *

- Single Source Shortest Path :
- Bellman - Ford Algorithm.



Updation :

$$\text{if } (d[u] + c[u,v] < d[v])$$

$$d[v] = d[u] + c[u,v]$$

exiting Point : ① if n-1 times iteration complete

② Convergence - if value of ~~2 Iterations is same~~. all nodes remain same in 2 iterat

(1,2) (1,3) (1,4) (2,5) (3,2) (3,5) (4,3) (4,6) (5,7) (6,7)

+ Iterations = (no. of vertices - 1)

1st Iteration — updation.

2nd " —

3rd Iteration —

4th Iteration } -> Same values (Exit Point)

Final weights:

	Shows the cost from
① → 0	1 - 1
2 → 1	1 - 2
3 → 3	1 - 3
4 → 5	1 - 4
5 → 0	1 - 5
6 → 4	1 - 6
7 → 3	1 - 7

Time complexity:

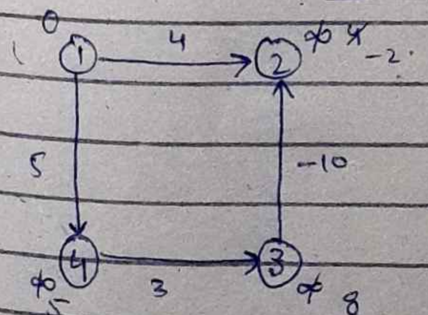
$$O(\overset{\text{vertices}}{|V|}\,\overset{\text{Edges.}}{|E|}$$

$$O(n \cdot n) = O(n^2).$$ In complete Graph.

Complete Graph means all possible edges. So in complete ~~Ed~~ Graph Edges will be $\frac{n(n-1)}{2}$ so,

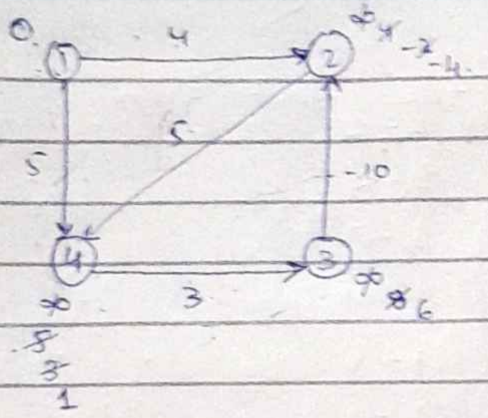
$$O\left(|V|\,|E|\right)$$

$$O\left(n \cdot \frac{n(n-1)}{2}\right) = n^3.$$ Complete Graph.



(1,2) (1,4) (3,2) (4,3)

Converges at iteration 3.



(1,2) (1,4) (3,2) (4,3) (2,4

Total iteration we have are 3 but it does not converges at 3rd iteration.

It updates at 4th iteration.

Alogrithm Failed (Draw Back).

Negative weight Cycle.

5 + 3 - 10 = -2.