LECTURE # 19

ADA

22-Dec-2023

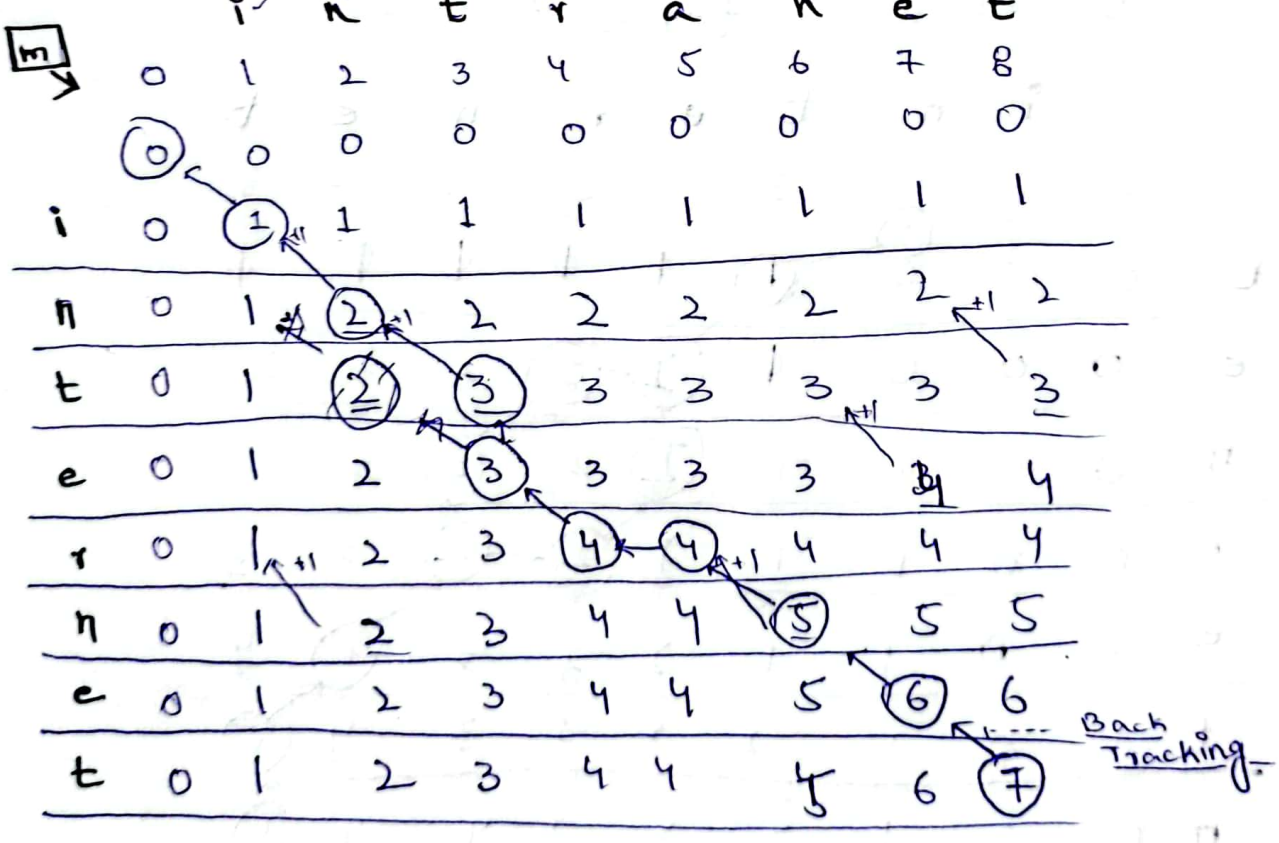
Friday

→ Dynamic Programming

- Identify and solve subproblems.
- Use Optimal substructure to construct/build the solution to the larger problem.
  - Local Optimal solution of the smaller subproblems.

Longest common Subsequence

- if letters doesn't match $m(i-1, j-1)+1$
- if " match chk for greater.



Length = 7 of Longest common Subsequence

→ internet
→ intranet
> intnet

Formula → if (A[i] = B[i])

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

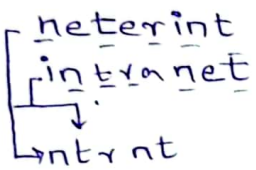
else

MAT[i,j] = max { MAT[i-1, j] , MAT[i, j-1] }

Time Complexity :-

O(n * m)

Example #1



→ Verify at home

Example #2

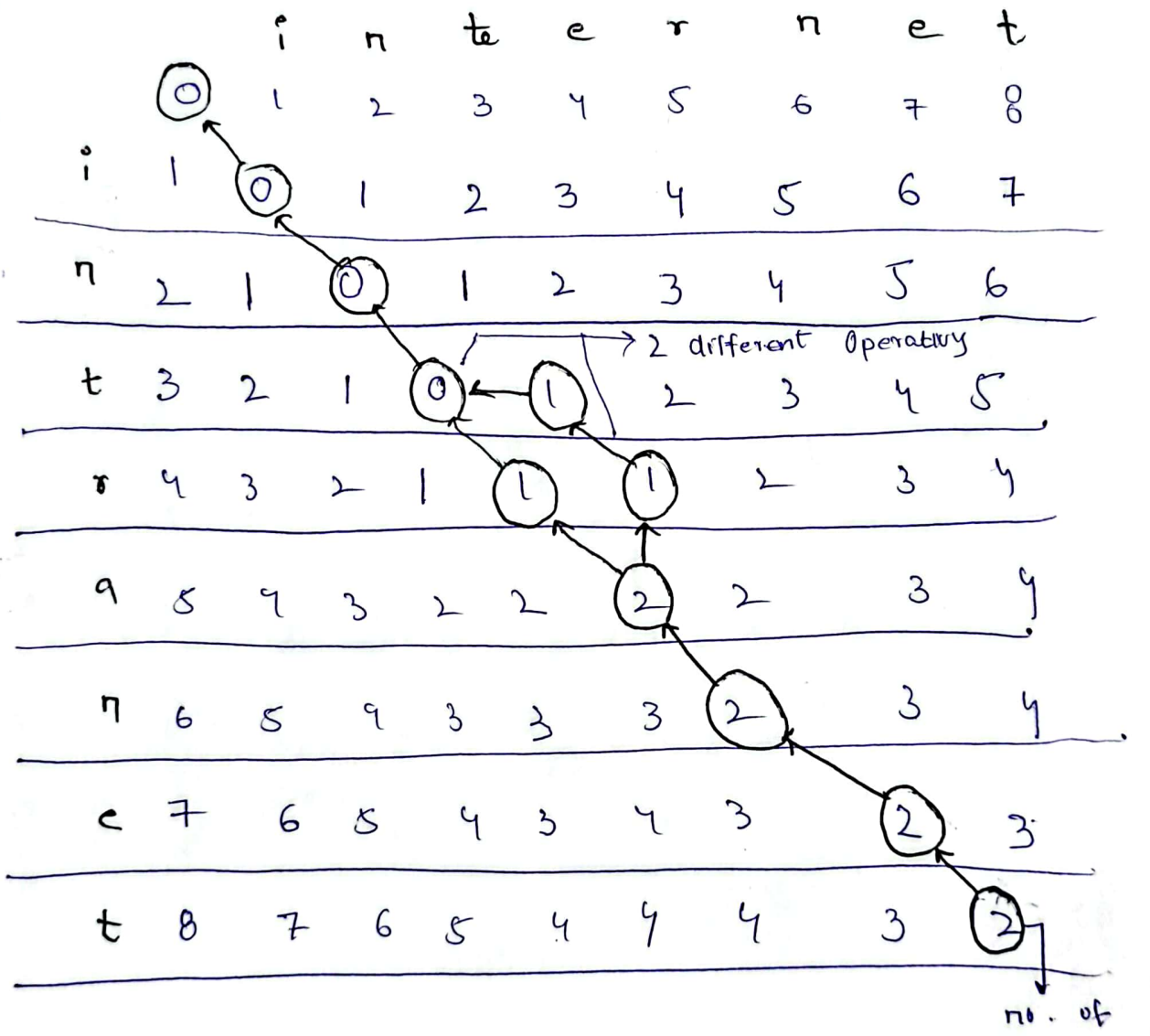
through backtracking

in ternet → teret
tenretni → tenet → length = 5

		i	n	t	e	r	n	e	t
	0	1	2	3	4	5	6	7	8
	0	0	0	0	0	0	0	0	0
t	0	0	0	ϕ	ϕ	ϕ	ϕ	ϕ	ϕ
e	0	0	0	ϕ	1	2	2	2	2
n	0	0	1	1	2	2	3	3	3
r	0	0	1	1	1	2	3	3	3
e	0	0	1	1	2	3	3	4	4
t	0	0	1	2	2	3	3	4	5
n	0	0	1	2	2	3	4	4	5
i	0	1	1	2	2	3	4	4	5

insert →
delete →

* Minimum Edit Distance



match = no operation

inter net
in tr a net