TEMPLATE MATCHING

- The Goal: Given a set of reference patterns known as TEMPLATES, find to which one an unknown pattern matches best. That is, each class is represented by a single typical pattern.

- The crucial point is to adopt an appropriate "measure" to quantify similarity or matching.

- These measures must accommodate, in an efficient way, deviations between the template and the test pattern. For example the word beauty may have been read a beeauty or beuty, etc., due to errors.

P1: 3.5, 0.9, 0.1 G

P2: 2.5 0.6, 0.4 B

P3: 3.8, 1.0, 0.0 G

Reference Patterns

Unknown Pattern

Pu: 2.9, 0.3, 0.5 ? P

Pu vs P1, 0.6, 0.6, 0.4

Pu vs P2 0.4, 0.3, 0.1

Pu vs P3 0.9, 0.7, 0.5

- Typical Applications
  - Speech Recognition
  - Motion Estimation in Video Coding
  - Data Base Image Retrieval
  - Written Word Recognition
  - Bioinformatis

- Measures based on optimal path searching techniques
  - Representation: Represent the template by a sequence of measurement vectors

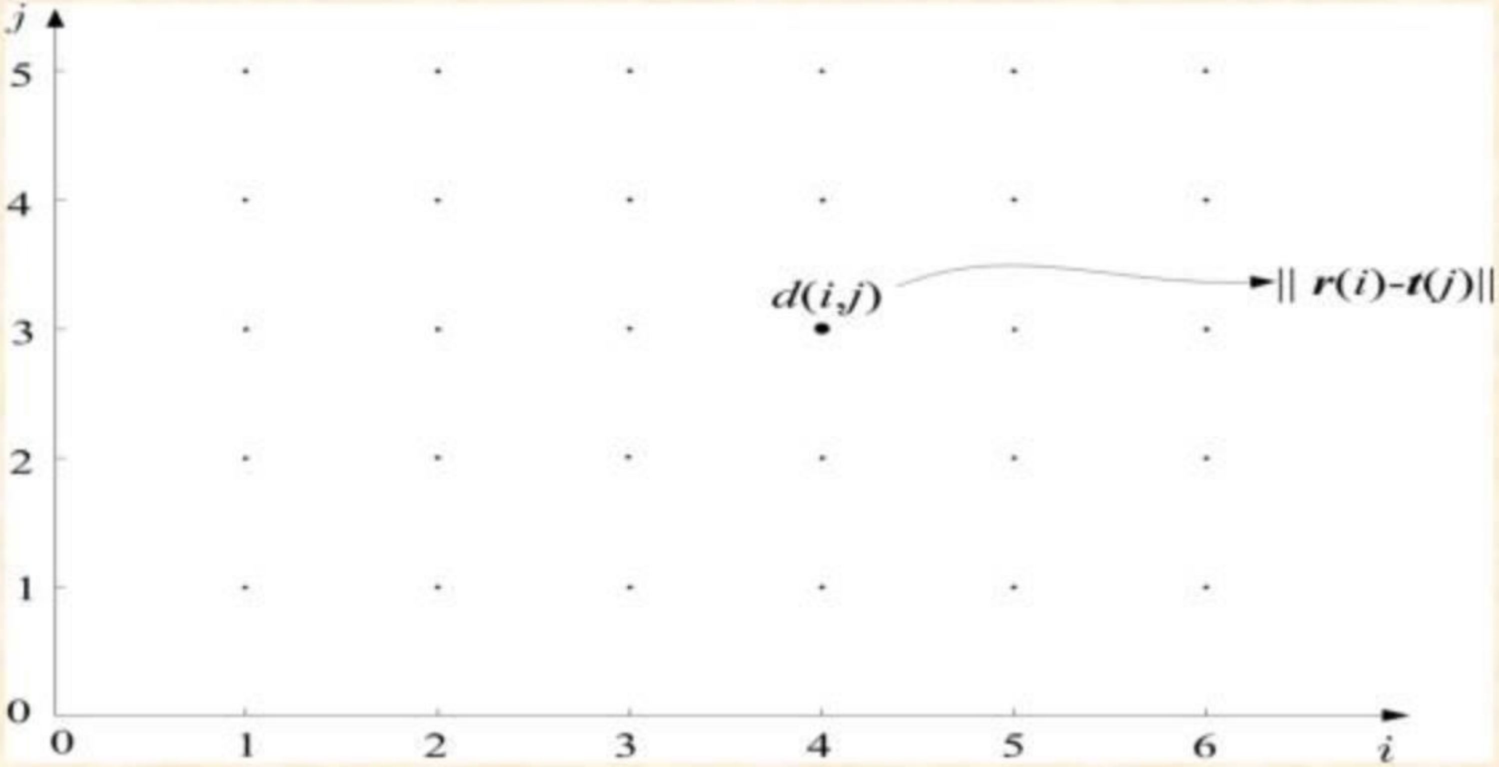
Template: $\underline{r}(1), \underline{r}(2), \ldots, \underline{r}(I)$

Test pattern: $\underline{t}(1), \underline{t}(2), \ldots, \underline{t}(J)$

- In general $I \neq J$

- Form a grid with I points (template) in horizontal and J points (test) in vertical

- Each point (i,j) of the grid measures the distance between $\underline{r}(i)$ and $\underline{t}(j)$

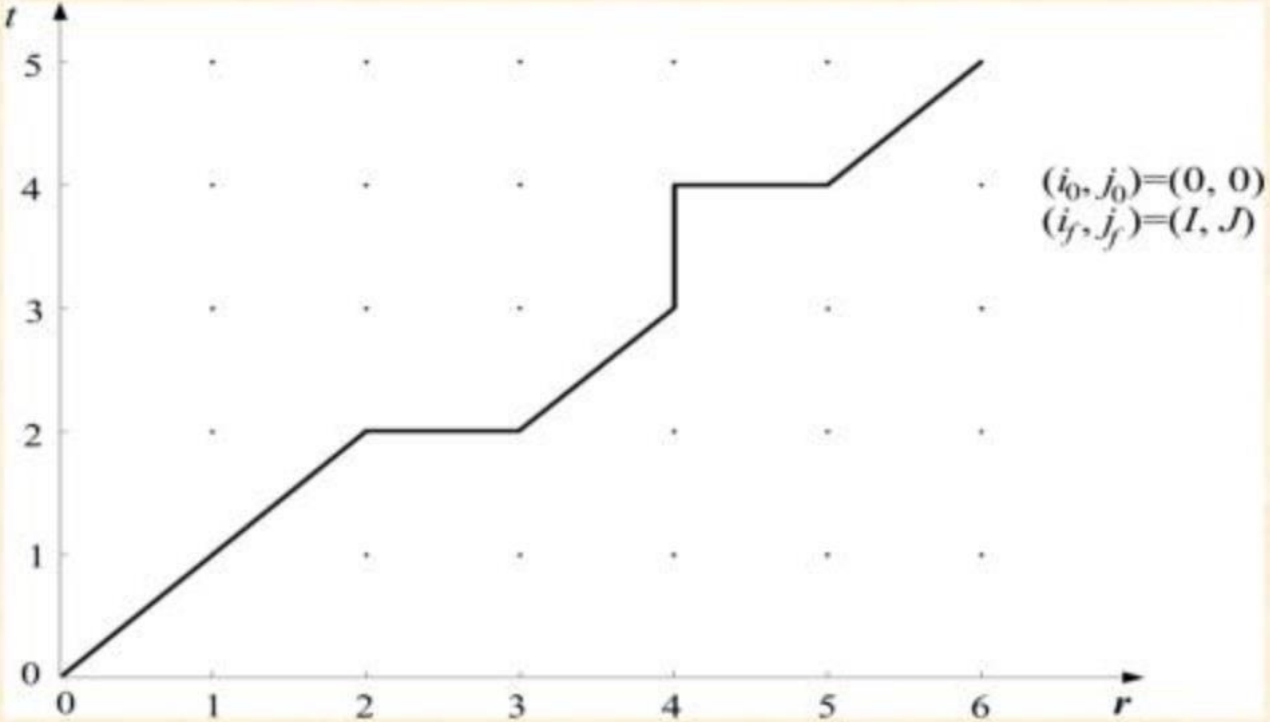


- **Path**: A path through the grid, from an initial node (i_0, j_0) to a final one (i_f, j_f), is an ordered set of nodes $(i_0, j_0), (i_1, j_1), (i_2, j_2) \dots (i_k, j_k) \dots (i_f, j_f)$

- Each path is associated with a cost

$$D = \sum_{k=0}^{K-1} d(i_k, j_k)$$

where K is the number of nodes across the path



- Search for the path with the optimal cost $D_{opt.}$
- The matching cost between template $\underline{r}$ and test pattern $\underline{t}$ is $D_{opt.}$

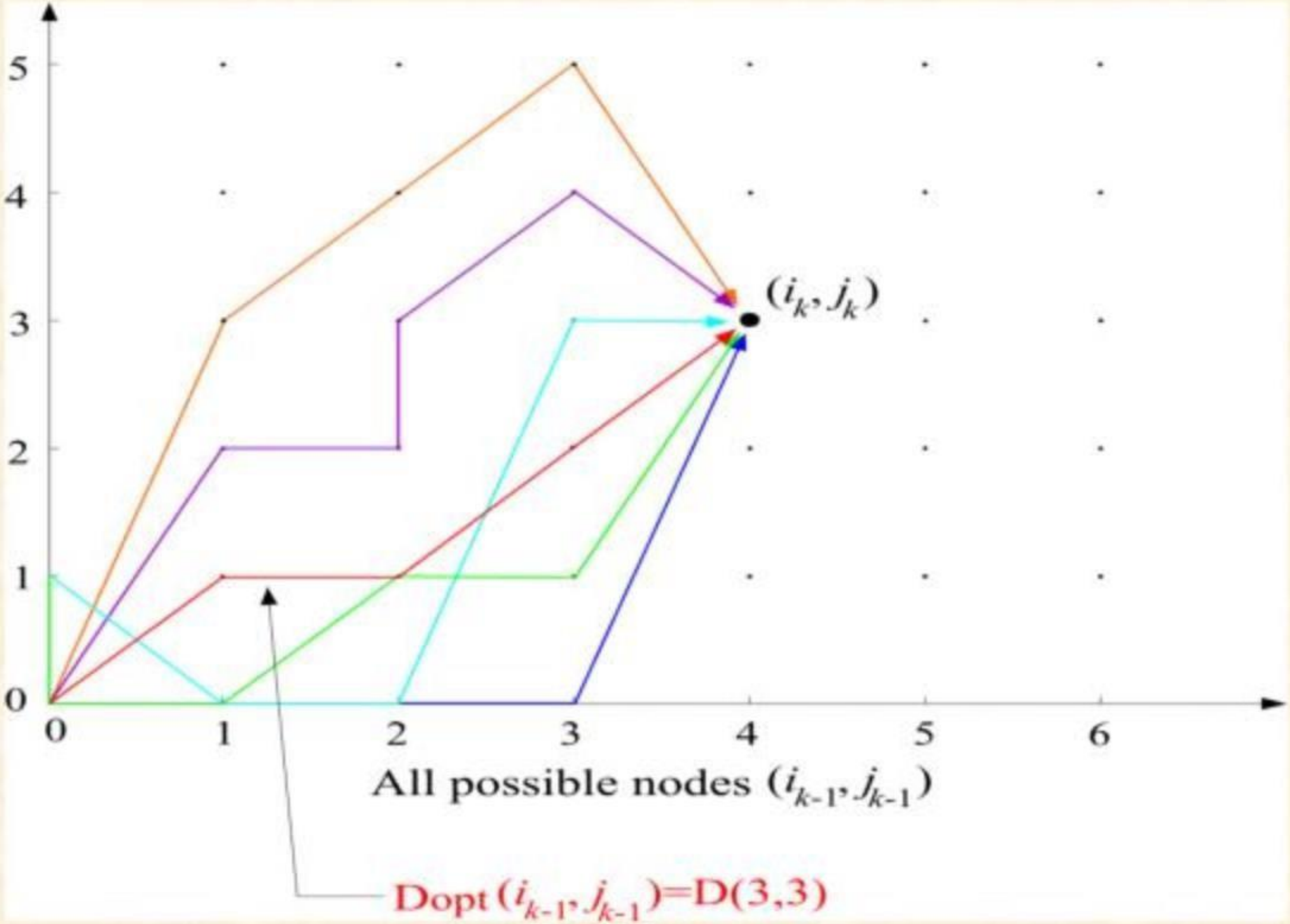
- Bellman's Principle:

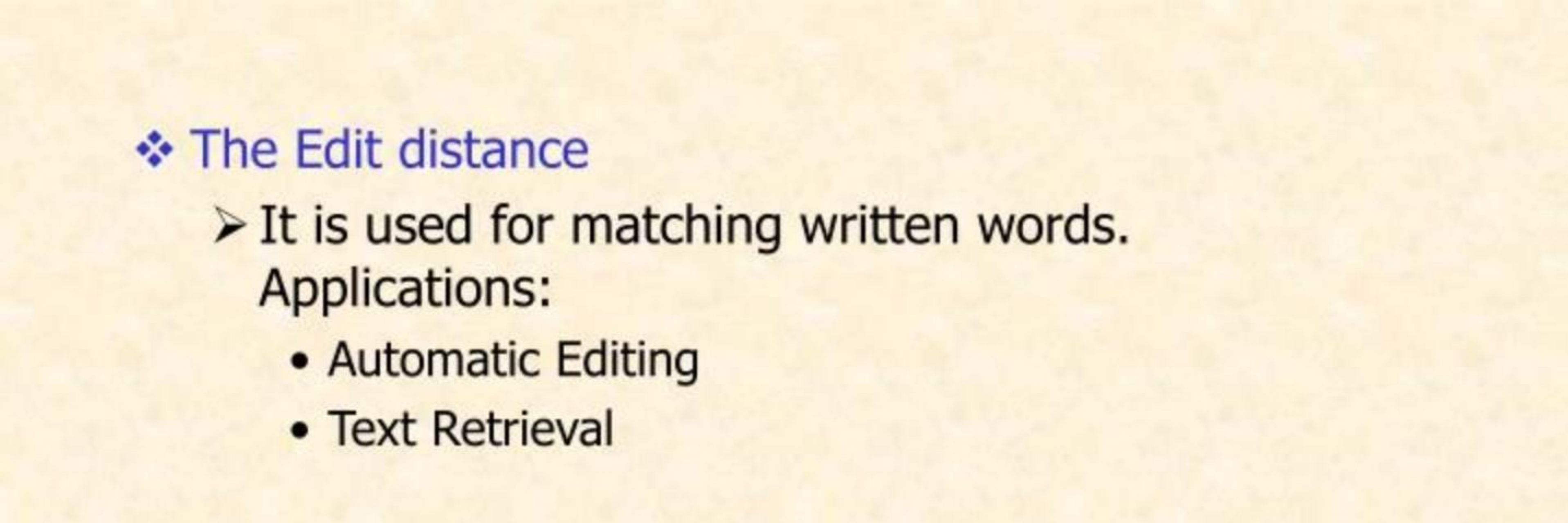
$$(i_0, j_0) \xrightarrow{opt} (i_f, j_f) = (i_0, j_0) \xrightarrow{opt} (i, j) \oplus$$
$$(i, j) \xrightarrow{opt} (i_f, j_f)$$

- In words: The overall optimal path from (i_0,j_0) to (i_f,j_f) through (i,j) is the concatenation of the optimal paths from (i_0,j_0) to (i,j) and from (i,j) to (i_f,j_f)

- Let $D_{opt.}\,(i,j)$ is the optimal path to reach (i,j) from (i_0,j_0), then Bellman's principle is stated as:

$$D_{opt}(i_k, j_k) = opt\{D_{opt}(i_{k-1}, j_{k-1}) + d(i_k, j_k)\}$$


5
4
3
2
1
0
0
1
2
3
4
5
6
(i_k, j_k)
All possible nodes (i_{k-1}, j_{k-1})
Dopt (i_{k-1}, j_{k-1})=D(3,3)



- ❖ The Edit distance
  - ➢ It is used for matching written words. Applications:
    - Automatic Editing
    - Text Retrieval

- The cost is based on the philosophy behind the so-called variational similarity, i.e.,
  - Measure the cost associated with converting one pattern to the other

- Edit distance: **Minimal** total number of changes, *C*, insertions I and deletions R, required to change pattern A into pattern B,

$$D(A,B)=\min_j [C(j)+I(j)+R(j)]$$

where j runs over All possible variations of symbols, in order to convert $A \longrightarrow B$

❖ Allowable predecessors and costs

- $(i-1, j-1) \rightarrow (i, j)$

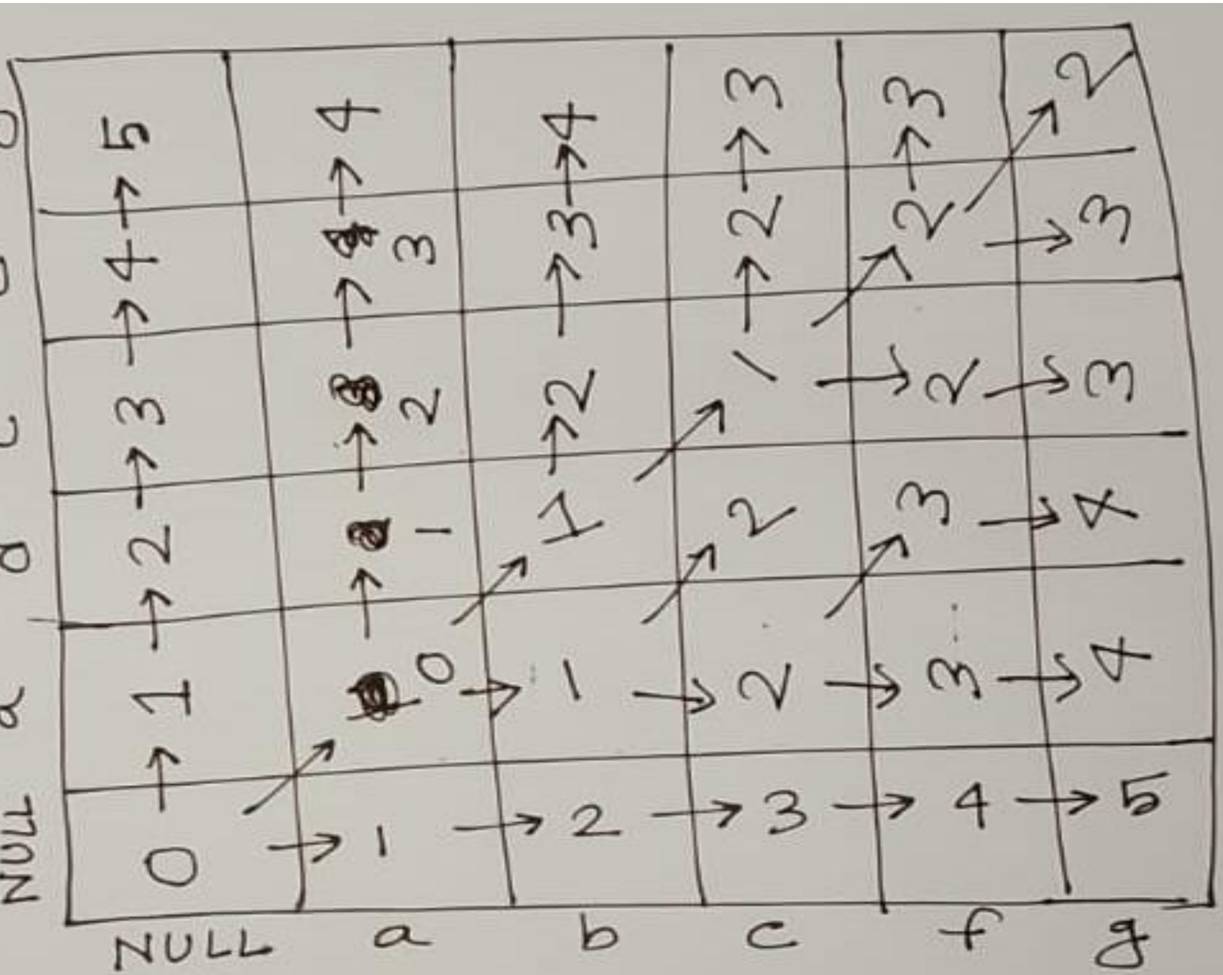
$$d(i, j \,|\, i-1, j-1) = \begin{cases} 0, & \text{if } t(i) = r(j) \\ 1, & t(i) \neq r(j) \end{cases}$$

- Horizontal (insert)

$$d(i, j \,|\, i-1, j) = 1$$

- Vertical (Delete)

$$d(i, j \,|\, i, j-1) = 1$$



g	5	4	4	3	3	2
e	4	3	3	2	2	3
c	3	2	2	1	2	3
d	2	1	1	2	3	4
a	1	0	1	2	3	4
NULL	0	1	2	3	4	5
	NULL	a	b	c	f	g

j
5
4
3
2
1
0

i						
	0	1	2	3	4	5

Question

Using the Edit Distance algorithm, find the minimum edit distance between the words “**intention**” and “**execution**”.

Show all steps and the final DP table.

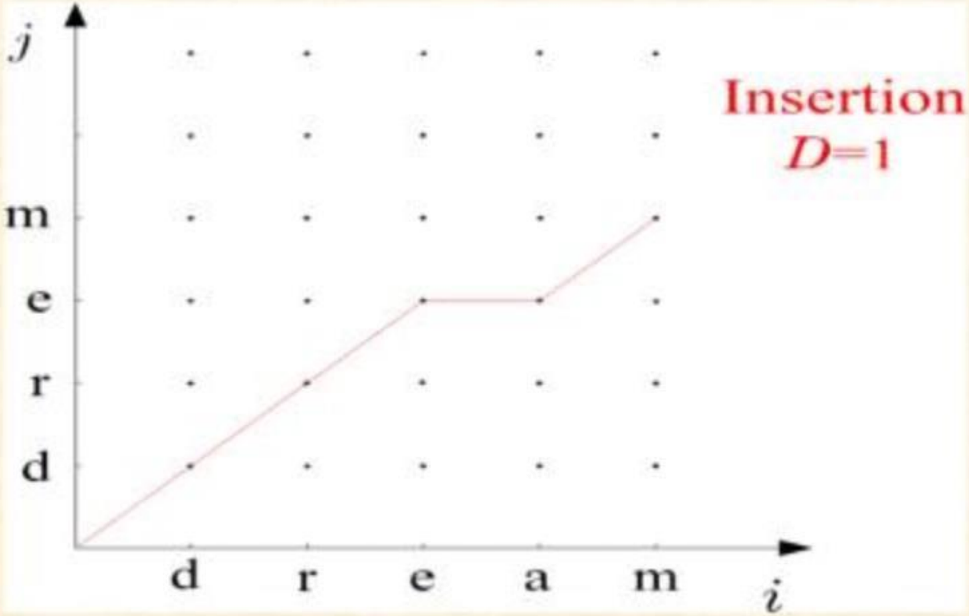
The Algorithm

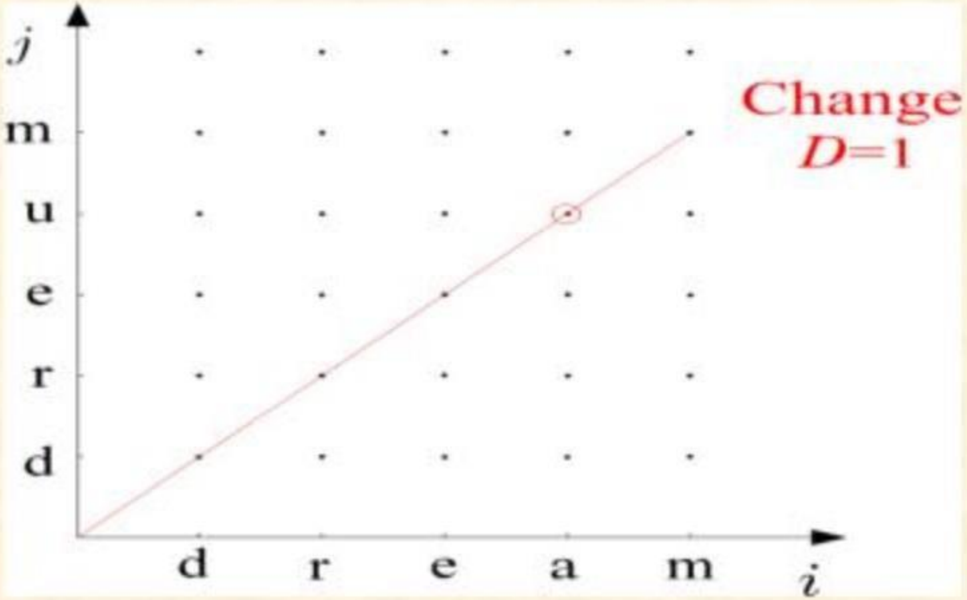
```
$D(0,0)=0$
For $i=1$, to $I$
    $D(i,0)=D(i-1,0)+1$
END {FOR}
For $j=1$ to $J$
    $D(0,j)=D(0,j-1)+1$
END{FOR}
For $i=1$ to $I$
    For $j=1$, to $J$
        $C_1=D(i-1,j-1)+d(i,j \mid i-1,j-1)$
        $C_2=D(i-1,j)+1$
        $C_3=D(i,j-1)+1$
        $D(i,j)=\min(C_1,C_2,C_3)$
    END {FOR}
END {FOR}
$D(A,B)=D(I,J)$
```

Question

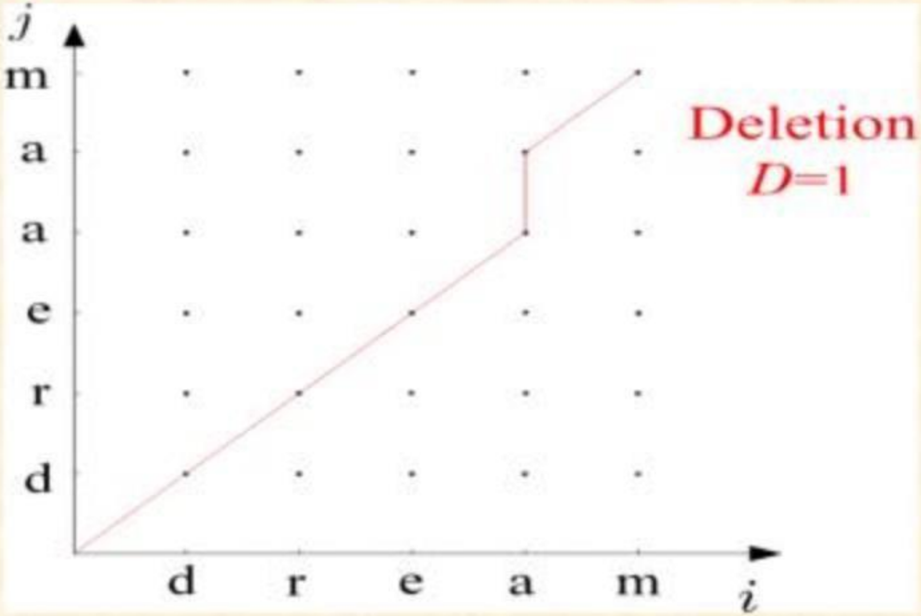
Using the edit distance algorithm, draw the paths for matching the three words **‘drem’**, **‘dreaam’,‘dreum’** to the actual word **‘dream’.**

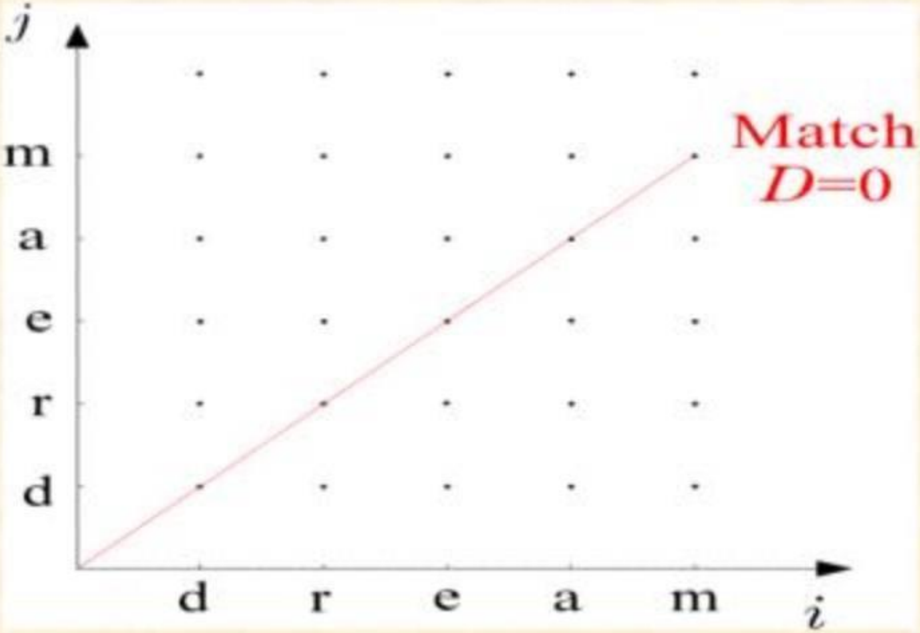
❖ Examples:





❖ Examples:


j
m
a
a
e
r
d
d
r
e
a
m
i
Deletion
D=1


j
m
a
e
r
d
d
r
e
a
m
i
Match
D=0