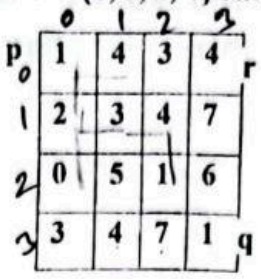
# Chapter 2: Digital Image Fundamentals

## *Origin42*

**Question 3. [Marks: 13]** [5]

b) Consider the image below with V = {0, 1, 2, 4} and answer the followings: [8]




i. What are the properties of Distance Function for Distance Metrics if there are 3 pixels p, q and r? [2]
ii. Calculate the distances: $D_4(p,q)$, $D_4(p,r)$, $D_8(p,q)$ and $D_e(p,q)$. [2]
iii. Define the terms $N_4(p)$, $N_D(p)$ and $N_8(p)$ of above image. [1.5]
iv. Does 4-path exist between p and q? Explain your answer. [1]
v. Does m-path exist between p and q? Explain your answer. [1.5]

**3.b. Solution: 024, Correction: 027**

**(i)**

Properties of Distance function:

- D(p, q) ≥ 0
- D(p, q) = D(q, p)
- D(p, r) ≤ D(p, q) + D(q, r)

**(ii)**

*Index count will start from (1, 1), not (0, 0).*
Given,

| | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | 1(p) | 4 | 3 | 4(r) |
| 2 | 2 | 3 | 4 | 7 |
| 3 | 0 | 5 | 1 | 6 |
| 4 | 3 | 4 | 7 | 1(q) |

D4 means City block distance. D4(p, q) = |1−4| + |1−4| = 6

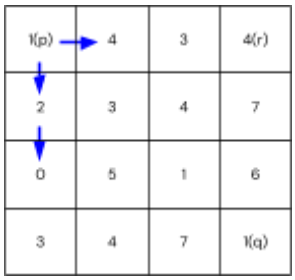
D4 means City block distance. D4(p, r) = |1-4| + |1-1| = 3
D8 means Chessboard distance. D8(p, q) = max(|1-4|, |1-4|) = 3
De means Euclidean distance. De(p, q) = sqrt((1-4)^2 + (1-4)^2) = 3sqrt(2)

**(iii)**

N4(p) means 4 neighbors of p. Their positions are top, down, left, right. For P they are 4, 2.
ND(p) means diagonal neighbors of p. They are 3.
N8(p) means 4 neighbors and diagonal neighbors of p. They are 4, 2, 3.

**(iv)**

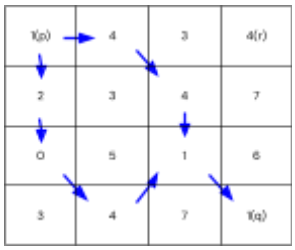
No. 4-path doesn't exist between p and q.




As, v={0, 1, 2, 4}. There is no 4-path from p to q containing values of v.

**(v)**

Yes. m-path exists between p and q.




As, v={0, 1, 2, 4}. There is an m-path from p to q containing values of v.

## *Enigma41*

**Question 4:**

b) Consider the two image subsets S1 and S2 and answer the following questions based on these two image subsets. [6]

| S1 | | | | | S2 | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 3 | 2 | 2 | 1(y) | 2 | 1 | 2 |
| 0 | 1 | 0 | 2 | 1 | 1 | 1 | 2 | 2 | 0 |
| 0 | 0 | 0 | 1 | 1 | 1 | 0 | 2 | 3 | 2 |
| 0 | 0 | 1(x) | 0 | 0 | 2 | 2 | 3 | 2 | 2 |
| 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 |

i. Draw a digital path from S1 or S2 and explain? [2]
ii. Does a four path exist between (x) and (y) for V = {1, 2}? [1]
iii. Do four path and m-path exists between (x) and (y) for V = {1, 0}? [2]
[If your answer is yes, draw the path. If no, explain why.]

iv. Determine whether S1 and S2 are 4-adjacent for V={1,2}? [1]

**4.b. Solution:**

## *Recursive40*

b) Consider the image segment shown on the table where V be the set of gray level values used to define the connectivity in the image. Compute $D_4$, $D_8$ and $D_m$ distances between pixel P and Q for, [6]

i. V= {2, 3}
ii. V= {2, 6}

| | | | | |
|---|---|---|---|---|
| 2(P) | 3 | 2 | 6 | 1 |
| 6 | 2 | 3 | 6 | 2 |
| 5 | 3 | 2 | 3 | 5 |
| 2 | 4 | 3 | 5 | 2 |
| 4 | 5 | 2 | 3 | 6(Q) |

**7.b. Solution:**
**(i)**

## *Prototype39*

**4.**

**a)** Consider the following matrix, which is part of an image. **[6]**

| | | |
|---|---|---|
| $q$ | $w$ | |
| | $p$ | |
| | $u$ | |

The values of pixels p, q, u and w belong to set V, which is the set of intensity values used to define adjacency.

i. Are pixels p and q 4-adjacent? Are they 8-adjacent? Are they m-adjacent? Provide one answer to each question.

ii. What is the Euclidean distance, the chessboard distance and the city-block distance between pixels q and u?

iii. Is there a 4-path from pixel u to pixel q? If so, what is it?

**Solution: 024**

**(i)**

p and q 4 adjacent: No
p and q 8 adjacent: Yes
p and q m adjacent: No

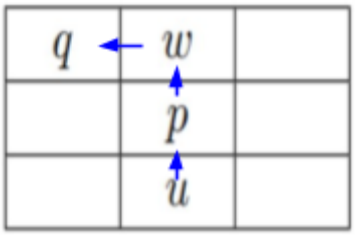
**(ii)**

| | 1 | 2 | 3 |
|---|---|---|---|
| 1 | $q$ | $w$ | |
| 2 | | $p$ | |
| 3 | | $u$ | |

So, q(1, 1) and u(3, 2).
De/Euclidean distance. De(q, u) = sqrt((1-3)^2 + (1-2)^2) = sqrt(5)
D8/Chessboard distance. D8(q, u) = max(|1-3|, |1-2|) = 2
D4/City block distance. D4(q, u) = |1-3| + |1-2| = 3

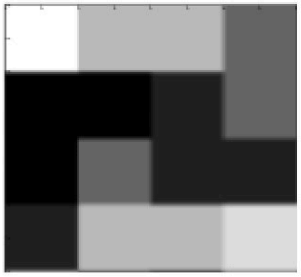
**(iii)**

Yes. 4-path exists between u and q.




As, v={p, q, u, w}. There is a 4-path from u to q containing values of v.

| | | |
|---|---|---|
| **b)** | Consider the following 4 × 4 image (at the left) with respective pixel values (at the right). Zoom it to a 6 × 6 image using nearest neighbor interpolation. | **[6.5]** |



| | | | |
|---|---|---|---|
| 180 | 160 | 160 | 140 |
| 110 | 110 | 120 | 140 |
| 110 | 140 | 120 | 120 |
| 120 | 160 | 160 | 170 |

**Solution: 024**
**Nearest Neighbor Interpolation:**
Given pixel values,

| | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | 180 | 160 | 160 | 140 |
| 2 | 110 | 110 | 120 | 140 |
| 3 | 110 | 140 | 120 | 120 |
| 4 | 120 | 160 | 160 | 170 |

Input image row=4, column=4.
Output image row=6, column=6.

Row ratio = 4/6, Column ratio = 4/6.

New row location = Row index x Row ratio
= {1 2 3 4 5 6} x (4/6)
= {0.67 1.33 2 2.67 3.33 4}
= {1 2 2 3 4 4} // taking ceiling values

Row interpolation values,

| | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | 180 | 160 | 160 | 140 |
| 2 | 110 | 110 | 120 | 140 |
| 2 | 110 | 110 | 120 | 140 |
| 3 | 110 | 140 | 120 | 120 |
| 4 | 120 | 160 | 160 | 170 |
| 4 | 120 | 160 | 160 | 170 |

New column location = Column index x Column ratio
= {1 2 3 4 5 6} x (4/6)
= {0.67 1.33 2 2.67 3.33 4}
= {1 2 2 3 4 4} // taking ceiling values

Column interpolation values,

| | 1 | 2 | 2 | 3 | 4 | 4 |
|---|---|---|---|---|---|---|
| 1 | 180 | 160 | 160 | 160 | 140 | 140 |
| 2 | 110 | 110 | 110 | 120 | 140 | 140 |
| 2 | 110 | 110 | 110 | 120 | 140 | 140 |
| 3 | 110 | 140 | 140 | 120 | 120 | 120 |
| 4 | 120 | 160 | 160 | 160 | 170 | 170 |
| 4 | 120 | 160 | 160 | 160 | 170 | 170 |

After row and column interpolation we get 6x6 pixel values