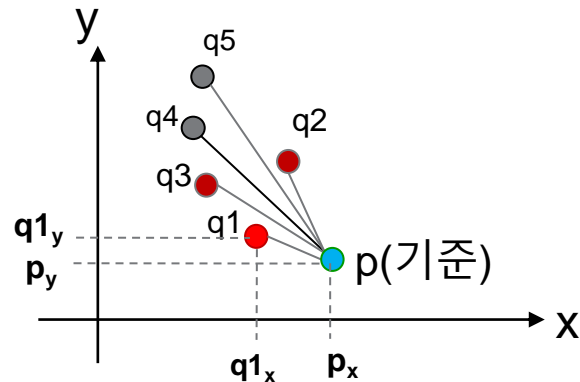


● k 최근접이웃(kNN : k Nearest Neighbor) 알고리즘

✓ 유클리드 거리계산식을 이용한 최근접이웃 k개를 선정하는 알고리즘

예) k=3 일 때 최근접이웃으로 선정된 q 3개(q1, q2, q3)



예) p점과 q1점 사이 거리계산식

$$\text{거리} = \text{sqrt}((p_x - q1_x)^2 + (p_y - q1_y)^2)$$

● 유클리드 거리계산식

$$\sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \dots + (p_n - q_n)^2} = \sqrt{\sum_{i=1}^n (p_i - q_i)^2}$$

관측대상 p와 q의 값의
차가 작으면, 두 관측
대상은 유사하다고 정의
하는 식

클래스 과제

✓ K 최근접이웃 (kNN : k Nearest Neighbor) 알고리즘 구현

<조건1> 파이썬 클래스와 기초문법 이용(파이썬 수업내용 적용)

<조건2> 데이터 생성 : random 모듈의 random() 함수 이용(0과 1 사이의 난수)

- 적용데이터 : $q[[x_1, y_1], [x_2, y_2], \dots [x_n, y_n]]$ → 10개(중첩 list)
- 기준데이터 : $p[x, y]$ → 1개(단일 list)

<조건3> 알고리즘 구현 내용

1. 거리(distance)계산 : 기준데이터(p)와 적용데이터(q) 간의 유클리드 거리계산식([앞 페이지 참고](#)) 이용
2. k 최근접이웃 선정 : 최근접이웃 1개, 3개, 5개 선정
3. 최근접 이웃 선정 결과(아래 출력 예시 참고)

k1 -> [0.18133] → 최근접이웃 1개 거리[distance]

real data : [0.91311, 0.12316] → 적용데이터 중에서 가장 가까운 1개 포인트[x좌표, y좌표]

k3 -> [0.18133, 0.18540, 0.24802] → 최근접이웃 3개 거리[distance]

real data : [0.91311, 0.12316] [0.75420, 0.45281] [0.62317, 0.18049] → 가장 가까운 3개 포인트 [x, y]

k5 -> [0.18133, 0.18540, 0.24802, 0.45897, 0.59520] → 최근접이웃 5개 거리[distance]

real data : [0.91311, 0.12316] [0.75420, 0.45281] [0.62317, 0.18049] [0.8892, 0.7480] [0.24990, 0.28701]
→ 가장 가까운 5개 포인트[x, y]