pearson

# example code

|  |
| --- |
| # 가장 일반적인 퀵 정렬  def quick\_sort1(array, start, end):      if start >= end: return # 원소가 1개인 경우      pivot = array[start] # 피벗은 첫 요소      left, right = start + 1, end        while left <= right:          # 피벗보다 작은 데이터를 찾을 때까지 반복          while left <= end and array[left] <= array[pivot]:              left += 1          # 피벗보다 큰 데이터를 찾을 때까지 반복          while right > start and array[right] >= array[pivot]:              right -= 1          if left > right: # 엇갈린 경우              array[right], array[pivot] = array[pivot], array[right]          else: # 엇갈리지 않은 경우              array[right], array[left] = array[left], array[right]      # 분할 이후 왼쪽 부분과 오른쪽 부분에서 각각 정렬 수행      quick\_sort1(array, start, right - 1)      quick\_sort1(array, right + 1, end)    # 파이썬의 장점을 살린 퀵 정렬  def quick\_sort2(array):      # 리스트가 하나 이하의 원소를 가지면 종료      if len(array) <= 1: return array        pivot, tail = array[0], array[1:]        leftSide = [x for x in tail if x <= pivot]      rightSide = [x for x in tail if x > pivot]        return quick\_sort2(leftSide) + [pivot] + quick\_sort2(rightSide)  import random    a = random.sample(range(0,100),100) # 0부터 99까지의 범위중에 10개를 중복없이 뽑겠다.  print(a)  print()  quick\_sort1(a,0,len(a) - 1)  print(a)  print()  b = random.sample(range(0,100),100) # 0부터 99까지의 범위중에 10개를 중복없이 뽑겠다.  print(b)  print()  print(quick\_sort2(b))  print() |

# testing result

|  |
| --- |
| **[29, 99, 32, 61, 22, 80, 25, 21, 59, 23, 79, 45, 35, 81, 78, 94, 72, 92, 63, 70, 97, 31, 43, 96, 64, 38, 30, 33, 53, 16, 27, 28, 86, 34, 65, 95, 41, 77, 20, 56, 19, 13, 39, 90, 9, 68, 15, 7, 46, 8, 87, 0, 76, 40, 58, 10, 82, 60, 36, 75, 66, 2, 74, 24, 54, 47, 73, 88, 91, 83, 26, 42, 14, 98, 12, 55, 89, 44, 93, 67, 18, 4, 11, 6, 51, 17, 49, 3, 69, 52, 57, 5, 84, 1, 85, 48, 37, 50, 62, 71]**  **[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 92, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 40, 28, 29, 31, 35, 32, 33, 30, 34, 36, 38, 37, 39, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 76, 53, 52, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 65, 64, 66, 68, 67, 69, 70, 77, 72, 73, 74, 75, 71, 78, 79, 80, 81, 82, 83, 84, 17, 85, 86, 87, 88, 89, 90, 91, 93, 94, 95, 96, 97, 98, 99]**  **[80, 10, 76, 42, 19, 12, 47, 72, 79, 9, 98, 63, 22, 74, 55, 20, 21, 34, 58, 11, 24, 73, 90, 26, 29, 83, 93, 15, 96, 6, 44, 54, 85, 81, 37, 60, 46, 4, 68, 7, 59, 84, 3, 99, 50, 35, 30, 40, 49, 53, 48, 94, 31, 51, 71, 89, 97, 88, 36, 92, 8, 61, 38, 57, 18, 95, 41, 91, 56, 32, 78, 17, 0, 65, 75, 25, 87, 28, 13, 16, 82, 77, 64, 1, 52, 5, 70, 69, 86, 43, 66, 62, 23, 14, 2, 27, 67, 45, 33, 39]**  **[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99]** |