Assignment 1 – Merge sort

|  |  |
| --- | --- |
| Student ID | 2021270660 |
| Name: | 이지원 |

* Following code is **merge\_sort(A, left, right)** function.

|  |
| --- |
| def merge\_sort(A, left, right) :  # sort A[left..right] ascending order    if left<right :      mid = (left + right) // 2      merge\_sort(A, left, mid)      merge\_sort(A, mid + 1, right)      merge(A, left, mid, right) |

1. Design your algorithm with Pseudo code for merge function. (4 points)

|  |
| --- |
| Merge(A, left, mid, right) :  L = A[left...mid]  R = A[mid+1..right]  i=0, j=0  k = left  while i<#L, k<#R:  if L[i] < R[j]:  A[k] = L[i]  I += 1  Else:  A[k] = R[j]  J += 1  k+=1  While I < #L:  A[k] = L[i]  I += 1  k += 1  While j < #R:  A[k] = R[j]  J += 1  Left += 1 |

1. Implement your algorithm in Python and explain your algorithm. (Include code comments 코드에 주석을 작성하시오) (4 points)

|  |
| --- |
| // Python implementation  def merge\_sort(A, left, right): # sort A[left..right] ascending order  if left < right:  mid = (left + right) // 2  merge\_sort(A, left, mid)  merge\_sort(A, mid + 1, right)  merge(A, left, mid, right)  return A  def merge(A, left, mid, right):  # mid 인덱스를 기준으로 양쪽으로 분할  L = A[left:mid+1] # A[left..mid]  R = A[mid+1:right+1] # A[mid+1..right]  i = 0  j = 0  k = left  # L과 R을 비교하면서 작은 것부터 A 배열에 병합  while i < len(L) and j < len(R):  if L[i] <= R[j]:  A[k] = L[i]  i += 1  else:  A[k] = R[j]  j += 1  k += 1  # L 배열에 남은 요소가 있으면 병합  while i < len(L):  A[k] = L[i]  i += 1  k += 1  # R 배열에 남은 요소가 있으면 병합  while j < len(R):  A[k] = R[j]  j += 1  k += 1  # test  tmp = [9, 2, 1, 3, 8, 12]  print(merge\_sort(tmp, 0, len(tmp)-1)) |

1. Test your **merge function** with random test inputs for verification of correctness. (2 points)

|  |
| --- |
| 텍스트, 스크린샷, 폰트이(가) 표시된 사진  자동 생성된 설명 |