과제

장유선

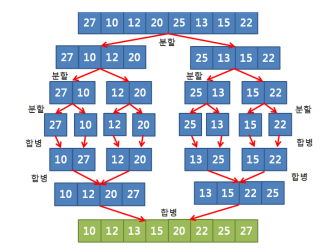
2023.08.28

1. 문제 정의

분할정복 알고리즘이었던 합병 정렬을 구현하고 0~1.000,000 범위 내에서 100,000개의 값을 추출하여 정렬 하세요.

2. 개념 설명

합병 정렬은 분할 정복 (Divide and Conquer) 방식을 이용해서 하나의 리스트를 두 개의 리스트로 분할한 다음 각각의 분할된 리스트를 정렬한 후에 합해서 정렬된 하나의 리스트로 만드는 정렬 알고리즘



분할(Divide) : 리스트를 두 개의 리스트로 분할한다

정복(Conquer) : 분할된 리스트를 정렬한다.

결합(Combine): 정렬된 두 개의 리스트를 하나의 정렬된 리스트로 결합한다.

3. Python Code Hard Copy

|  |
| --- |
| import random  def merge\_sort(lst):      if len(lst) <= 1:          return lst      mid = len(lst) // 2        left = merge\_sort(lst[:mid])      right = merge\_sort(lst[mid:])      return merge(left, right)  def merge(left, right):      result = []      i, j = 0, 0      while i < len(left) and j < len(right):          if left[i] <= right[j]:              result.append(left[i])              i += 1          else:              result.append(right[j])              j += 1      result += left[i:]      result += right[j:]      return result  random\_list = [random.randint(0, 1000000) for \_ in range(100000)]  sorted\_list = merge\_sort(random\_list)  print("정렬된 배열의 처음 10개 값:", sorted\_list[:10])  print("정렬된 배열의 마지막 10개 값:", sorted\_list[-10:]) |

4. Code 설명

4-1. 합병 정렬 함수

def merge\_sort(lst):

    if len(lst) <= 1:

        return lst

    mid = len(lst) // 2

    left = merge\_sort(lst[:mid])

    right = merge\_sort(lst[mid:])

    return merge(left, right)

* 리스트의 길이가 1 이하면 이미 정렬되어 있는 상태 -> 반환
* 중간지점을 기준으로 왼쪽 부분과 오른쪽 부분을 재귀적으로 정렬한다.
* 정렬된 왼쪽과 오른쪽 부분을 merge 함수를 이용해 합병한다.

4-2. 합병 함수

def merge(left, right):

    result = []

    i, j = 0, 0

    while i < len(left) and j < len(right):

        if left[i] <= right[j]:

            result.append(left[i])

            i += 1

        else:

            result.append(right[j])

            j += 1

    result += left[i:]

    result += right[j:]

    return result

* 왼쪽 부분 리스트의 값이 오른쪽 부분 리스트의 값보다 작거나 같으면, result.append(left[i])를 실행하여 결과 리스트에 왼쪽 부분 리스트의 값을 추가하고, i += 1을 실행하여 다음 값을 처리
* 왼쪽 부분 리스트의 값이 오른쪽 부분 리스트의 값보다 크면, result.append(right[j])를 실행하여 결과 리스트에 오른쪽 부분 리스트의 값을 추가하고, j += 1을 실행하여 다음 값을 처리
* 결과 리스트에 리스트의 남은 값을 추가

5. 결과

|  |
| --- |
| 정렬된 배열의 처음 10개 값: [0, 2, 13, 19, 19, 31, 35, 39, 40, 93]  정렬된 배열의 마지막 10개 값: [999911, 999915, 999920, 999933, 999935, 999936, 999955, 999972, 999980, 999992] |

6. 결과 화면

