

〈알고리즘 실습〉 - 합병정렬

※ 입출력에 대한 안내

- 특별한 언급이 없으면 문제의 조건에 맞지 않는 입력은 입력되지 않는다고 가정하라.
- 특별한 언급이 없으면, 각 줄의 맨 앞과 맨 뒤에는 공백을 출력하지 않는다.
- 출력 예시에서 □는 각 줄의 맨 앞과 맨 뒤에 출력되는 공백을 의미한다.
- 입출력 예시에서 ↦ 이 후는 각 입력과 출력에 대한 설명이다.

[문제 1] (합병 정렬) N개의 양의 정수를 입력(중복 허용)받아 정렬하는 프로그램을 작성하시오. 정렬은 단일연결리스트를 이용하여 합병정렬을 구현하여 사용한다.

○ 구현해야할 합병 정렬 알고리즘:

- 크기가 N인 단일연결리스트를 동적 할당하여, 입력된 양의 정수 저장 (입력 정수는 중복 허용)
- mergeSort(L) 함수: 단일연결리스트 L의 원소들을 합병정렬하여 정렬된 결과를 오름차순으로 정렬
- merge(L1, L2) 함수: mergeSort에 호출되어 두 개의 정렬된 단일연결리스트 L1과 L2를 합병한 하나의 단일연결리스트를 반환. 합병을 위해서 새로운 공간을 할당하면 안되고, L1과 L2 노드들의 링크만 변화시켜서 합병.
- mg-partition(L, k) 함수: 단일연결리스트 L과 양의 정수 k를 입력받아서 L을 크기가 k이고 |L|-k인 두 개의 부분리스트 L1과 L2로 분할하여 (L1, L2)를 반환. 여기서 |L|은 L의 크기. 분할 시에도 추가로 공간을 할당해서 사용하지 않고, L의 공간을 그대로 사용해서 분할.

입력 예시 1

출력 예시 1

3 ↦ N 4 9 1	□ 1 4 9 ↦ 정렬 결과
-----------------------------------	------------------------------------

입력 예시 2

출력 예시 2

8 ↦ N 73 65 48 31 29 20 8 3	□ 3 8 20 29 31 48 65 73 ↦ 정렬 결과
---	--

힌트: 다음은 합병 정렬의 배열 구현 알고리즘이다. 곧바로 연결리스트로 합병정렬을 구현하기 어려운 경우에는 아래의 배열 구현을 먼저하고, 연결리스트 구현을 해볼 것을 권장. 아래 배열 구현에서는 merge시에 B라는 보조적 배열공간을 추가적으로 만들어 사용하였으나, 문제1의 리스트구현에서는 추가로 공간을 사용하지 않는다.

<pre> Alg mergeSort(A) input array A of n keys output sorted array A 1. rMergeSort(A, 0, n - 1) 2. return Alg rMergeSort(A, l, r) input array A[l..r] output sorted array A[l..r] 1. if (l < r) m ← (l + r)/2 rMergeSort(A, l, m) rMergeSort(A, m + 1, r) merge(A, l, m, r) 2. return </pre>	<pre> Alg merge(A, l, m, r) input sorted array A[l..m], A[m+1..r] output sorted array A[l..r] merged from A[l..m] and A[m+1..r] 1. i, k ← l 2. j ← m + 1 3. while (i ≤ m & j ≤ r) if (A[i] ≤ A[j]) B[k++] ← A[i++] else B[k++] ← A[j++] 4. while (i ≤ m) B[k++] ← A[i++] 5. while (j ≤ r) B[k++] ← A[j++] 6. for k ← l to r A[k] ← B[k] 7. return </pre>
--	---