컴퓨터 알고리즘 4주차 과제

2019040164 정지오

1. Pseudo-code

MergeSort(list,left,right)

입력: list[left]~list[right]

출력: 정렬된 list[left]~list[right]

1. if ( left < right ) { // 배열의 원소의 수가 2개 이상이면

2. mid = ⌊(left+right)/2⌋ // mid=반으로 나누기 위한 중간 원소의 인덱스

3. MergeSort(list,left,mid) // 앞부분 재귀 호출

4. MergeSort(list,mid+1,right) // 뒷부분 재귀 호출

5. list[left]~list[mid]와 list[mid+1]~list[right]를 합병한다.

6. }

Merge(list, left, mid ,right)

(i = left; j = mid+1; k = left)

입력: list[left]~list[mid], list[mid+1]~list[right]

출력: 정렬된 하나의 list[left …. Right]

1. while(i<=mid && j<=right){ // i값이 mid값보다 작고 j값이 right값보다 작을때까지

2. if(list[i]<=list[j]) // i값이 j값보다 작으면

3. sorted[k++] = list[i++]; // i값을 sorted에 넣고 i,k 값을 하나씩 증가

4. else // 그렇지 않으면

5. sorted[k++] = list[j++]; // j값을 sorted에 넣고 j,k 값을 하나씩 증가

6. }

7. if(i>mid){ // 만약 i값이 미드보다 커지면(i는 다 옮기고 j만 남음)

8. for(l=j; l<=right; l++) // 현재 j위치를 l로 잡고 right값까지 l을 하나씩 증가

9. sorted[k++] = list[l]; // l값을 sorted에 넣고 k 값 하나 증가

10. }

11. else{ // 그렇지 않다면(반대의 경우)

12. for(l=i; l<=mid; l++) // 현재 i위치를 l로 잡고 mid값까지 l을 하나씩 증가

13. sorted[k++] = list[l]; // l값을 sorted에 넣고 k 값 하나씩 증가

14. }

15. for(l=left; l<=right; l++){ // l값은 left로 설정하고 right값까지 l값을 하나씩 증가

16. list[l] = sorted[l]; // sorted[l]을 list[l]값으로 저장

17. }

18. }

2. 코드

텍스트이(가) 표시된 사진

자동 생성된 설명텍스트이(가) 표시된 사진

자동 생성된 설명