HW9: Merge sort and heap sort

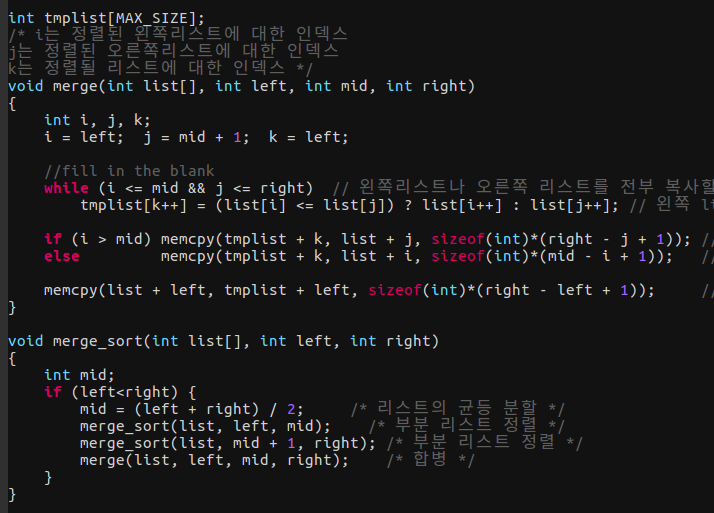
2012003567

이상화

**1. 문제 해결의 주요 코드**

**topic1. Merge Sort**

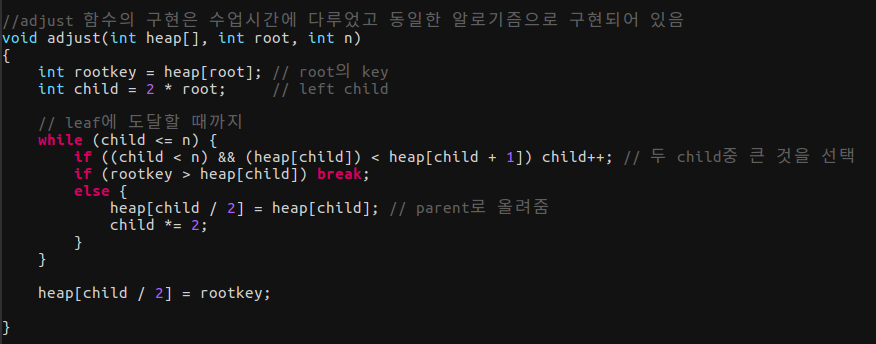
(1) void merge(int list[], int left, int mid, int right)



정렬된 list[left:mid]와 list[mid+1:right]를 합병하는 함수이다. 먼저 정렬한 list를 임시로 저장할 tmplist배열을 전역변수로 선언해주었다. while문에서 하는 작업은 왼쪽 list의 첫번째 index i(left)와 오른쪽 list의 첫번째 index j(mid+1)부터 비교해서 작은 값을 tmplist에 순차적으로 적어주는 것이다. 그다음 if,else문에서는 남은 list의 값들을 tmplist로 복사해 주는 작업을 한다. 복사는 memcpy 함수를 이용해서 했다. 먼저 i>mid인 경우는 오른쪽 list의 값들이 남은 상태이다. tmplist의 k번째 index부터 복사해야 하므로 dst값에 tmplist+k를 주었고 list의 j번째 index부터 남았기 때문에 src에 list+i의 값을 주었다. 4바이트만큼 right – j + 1만큼을 복사해야 하기 때문에 sizeof(int)\*(right-j+1)값을 세번째 인자로 전달했다. 왼쪽 list가 남은 상태는 j와 right값을 i와 mid값으로 바꿔주면 동일하다.

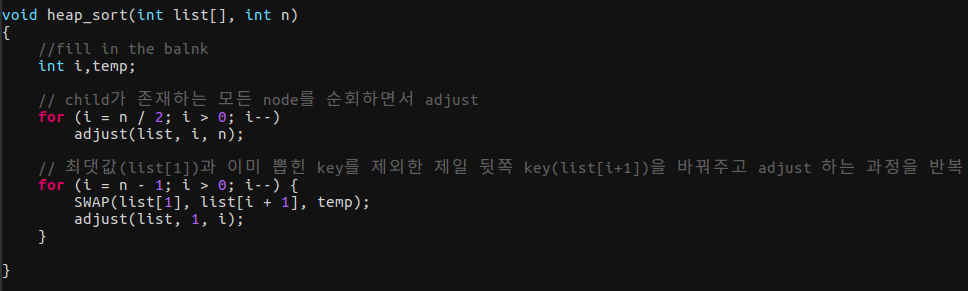
**topic2. Heap Sort**

(1) void adjust(int heap[], int root, int n)



adjust함수는 root의 left,right subtree는 모두 heap인 상태에서 전체 tree가 heap이 되도록 rootkey의 값을 조정해 주는 함수이다. while(child <= n)은 leaf에 도달 했을 때 탈출하기 위함이다. 첫번째 if문에서 left child와 right child를 비교해서 child 값을 더 큰 값의 index로 바꿔준다. 그 뒤 rootkey의 값과 child의 값을 비교해서 child가 더 크면 parent로 올려주는 작업을 반복한다. 이 때 rootkey가 child보다 더 크면 더이서 조정해줄 필요가 없으므로 while문을 탈출하고 child/2위치에 rootkey를 삽입한다.

(2) void heap\_sort(int list[], int n)



heap\_sort 함수의 첫번째 for문을 leaf부터 순차적으로 adjust함수를 적용해서 heap구조가 아니었던 list를 heap으로 만들어주는 과정이다. root의 index를 n/2부터 시작한 이유는 leaf일때는 child가 없으므로 고려해줄 필요가 없기 때문이다. 두 번째 for문에서 정렬을 완성한다. 이미 만들어진 heap에서 1번 index에 들어있는 값을 max값을 의미한다. 이 max값과 heap의 맨 마지막 index를 SWAP해준 뒤에 다시 adjust를 해주는 과정을 반복하면 list의 뒤쪽에는 max값이 순차적으로 쌓이게 되며 정렬이 완료된다.

**2. 실행결과**

