**알고리즘 과제2**

(합병정렬 반복적 프로그램 작성)

logo.gif

|  |  |
| --- | --- |
| 과목명 | 알고리즘 |
| 담당교수 | 김계영 |
| 학과 | 소프트웨어학부 |
| 학년 | 3 |
| 학번 | 20152994 |
| 성명 | 이진영 |
| 제출일 | 2017.10.10 |



**과제2 – 합병정렬 반복적 프로그램 작성**

20152994 이진영

**1. 원시 코드**

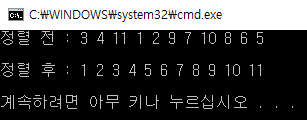
|  |
| --- |
| #include<stdio.h>  #include<malloc.h>  #define ARR\_SIZE 11  void mergeSort(int arr[], int num);  /\* arr을 이미 원소 1개 단위까지 분할된 배열로 보고 합병정렬 수행 (분할과정 생략) \*/  void mergeUnit(int arr[], int num, int unit);  /\* unit개 단위로 arr을 나누고 2개 단위씩을 묶어가며 merge() 함수 호출하여 arr 정렬 \*/  void merge(int arr[], int res\_arr[], int l\_first\_idx, int l\_last\_idx, int r\_last\_idx);  /\* arr을 left/right 두 부분으로 나눠 2개의 배열로 보고 merge하면서 정렬하여 res\_arr에 저장 \*/  void arrayCopy(const int src[], int dst[], int size); // 배열을 src에서 dst로 복사  int ex\_arr[ARR\_SIZE] = { 3,4,11,1,2,9,7,10,8,6,5 };  void main() {  printf("정렬 전 : ");  for (int i = 0; i < ARR\_SIZE; i++) {  printf("%d ", ex\_arr[i]);  }  printf("\n\n");  mergeSort(ex\_arr, ARR\_SIZE);  printf("정렬 후 : ");  for (int i = 0; i < ARR\_SIZE; i++) {  printf("%d ", ex\_arr[i]);  }  printf("\n\n");  }  void mergeSort(int arr[], int num) {  int unit = 1;  while (unit < num) {  mergeUnit(arr, num, unit);  unit \*= 2;  }  }  void mergeUnit(int arr[], int num, int unit) {  int i = 0, j;  int\* tmp\_arr;  tmp\_arr = (int\*)calloc(num, sizeof(int));  for (; i + 2 \* unit - 1 < num; i += (2 \* unit)) {  merge(arr, tmp\_arr, i, i + unit - 1, i + 2 \* unit - 1);  }  if (i + unit - 1 < num - 1) {  merge(arr, tmp\_arr, i, i + unit - 1, num - 1);  }  else {  for (j = i; j < num; j++)  tmp\_arr[j] = arr[j];  }  arrayCopy(tmp\_arr, arr, num);    free(tmp\_arr);  }  void merge(int arr[], int res\_arr[], int l\_first\_idx, int l\_last\_idx, int r\_last\_idx) {  int i = l\_first\_idx;  int j = l\_last\_idx + 1; // r\_first\_index  int k = i;  while (i <= l\_last\_idx && j <= r\_last\_idx) {  if (arr[i] <= arr[j])  res\_arr[k++] = arr[i++];  else  res\_arr[k++] = arr[j++];  }  if (i > l\_last\_idx) {  while (j <= r\_last\_idx)  res\_arr[k++] = arr[j++];  }  else {  while (i <= l\_last\_idx)  res\_arr[k++] = arr[i++];  }  }  void arrayCopy(const int src[], int dst[], int size) {  for (int i = 0; i < size; i++) {  dst[i] = src[i];  }  } |

**2. 실행 결과**

**① 입력 자료**

int ex\_arr[ARR\_SIZE] = { 3,4,11,1,2,9,7,10,8,6,5 };

**② 실행 결과**

****