2019년 알고리즘

- HW 06 -

제출일자	2019.10.15.
이 름	장수훈
학 번	201402414
분 반	01

Quick Sort

코드 설명

data06.txt 파일의 string 들을 읽어와 ,를 기준으로 스트링 배열에 넣어주었다. 이후 형변환을 통해 인트형 배열 array, array2에 넣어주었다. 배열 2개에 따로 넣은 이유는 랜덤한 킷값으로 정렬할 때 이미 정렬된 배열을 사용하면 안되서 그렇다.

```
public static void sort(int[] arr, int p, int r) {
                   if(p < r) {
   int q = partition(arr,p,r);</pre>
                                      sort(arr,p,q-1);
sort(arr,q+1,r);
  public static void quicksort_withrandom(int[] arr , int p , int r) {
                     int q = 0;
if(p < r) {</pre>
                                       q = randonized partition(arr,p,r);
                                      quicksort_withrandom(arr,p,q-1);
quicksort_withrandom(arr,q+1,r);
1
 public static int partition(int[] arr, int p, int r) {
                 lic static int per class
int x = arr[r];
int i = p-1;
for(int j = p; j <= r-1; j++) {
    if(arr[j] <= x) {
        i = i + 1;
        i = i + i;
        i = i 
                                                 swap(arr, i, j);
                                  }
                }
i = i +1;
swap(arr, i, r);
return i;
 public static void swap(int[] arr, int a, int b) {
                     int temp = arr[a];
arr[a] = arr[b];
arr[b] = temp;
 public static int randonized_partition(int[] arr, int p, int r){
                     Random rand= new Random();
int i = rand.nextInt(r-p) + p;
                   swap(arr, r, i);
return partition(arr,p,r);
```

코드 설명

sort 함수는 divide and conquer을 기반으로 하는 내부 정렬 알고리즘을 이용한 함수로서 분할, 정복, 결합을 수행하는 2개의 프로시저가 필요하다. partition 함수는 인자를 배열, p, r 로받아 (p,r,은 처음과 끝 인덱스) 정렬을 하여 {x}의 인덱스를 반환한다. arr[i] <= X를 조건 으로 참일 때 I를 증가시키고 I번째 인덱스 벨류와 i번쨰 인덱스 벨류를 교환한다. for문이 끝나면 I를 1증가시키고 I번쨰와 r 번쨰 벨류를 교환한다. 최종 리턴값은 x가 위치한 곳의 인덱스 I를 반환하고 이를 이용해 sort함수에서 재귀호출을 하게된다.

결과값

data, paids - 49.29
1980 | Aliver | 29.70 | 58.89
1980 | Aliver | 29.70 | 59.70 | 59.89
1980 | Aliver | 29.70 | 59.70 | 59.89
1980 | Aliver | 29.70 | 59.80
1980 | Aliver | 29.70
1980 | Aliver | 29.70
1980 | Alive