연습문제 12장 - 201910783 김성현

01. ④   38497

02. ②   67359

03. ①   1pass = 38497 / 2pass = 34897 / 3pass = 34789

04. ㉠ : j >= 0 && A[j] > key

㉡ : A[j+1] = A[j];

①

05. step 1

(1) 선택정렬 34967875 → 34967875 → 34567879 → 34567879 → 34567897

★ 34567789 ← 34567789

step 1

(2) 삽입정렬 47963875 → 47963875 → 46793875 → 34679875

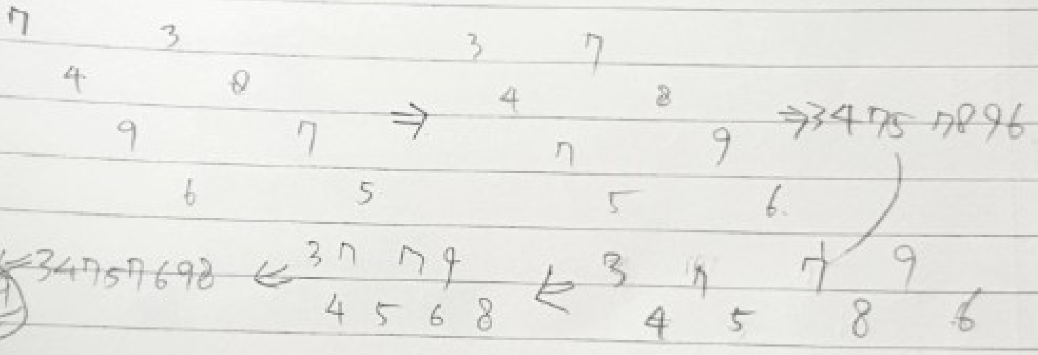
★ 34567789 ← 34677895 ← 34678975

step 1

(3) 버블정렬 47638759 → 46377589 → 43675789 → 34657789

★ 34567789 ← 34567789 ← 34567789 ← 34567789

★(4) 쉘정렬



06. (1) 퀵정렬.

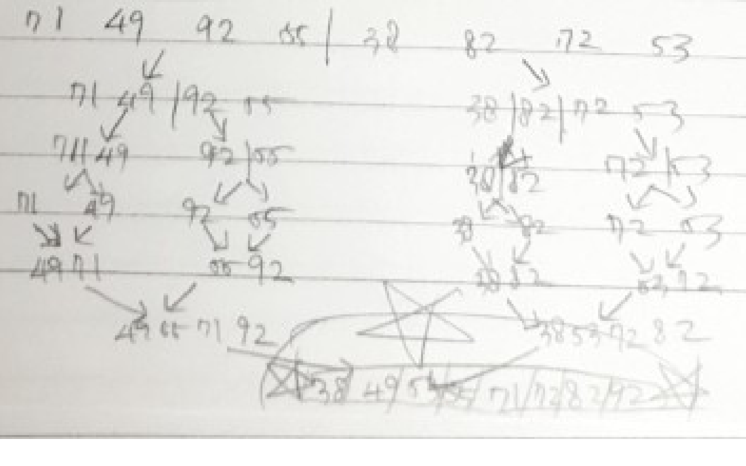
71 49 92 55 38 82 72 53

53 49 55 38 71 82 72 92

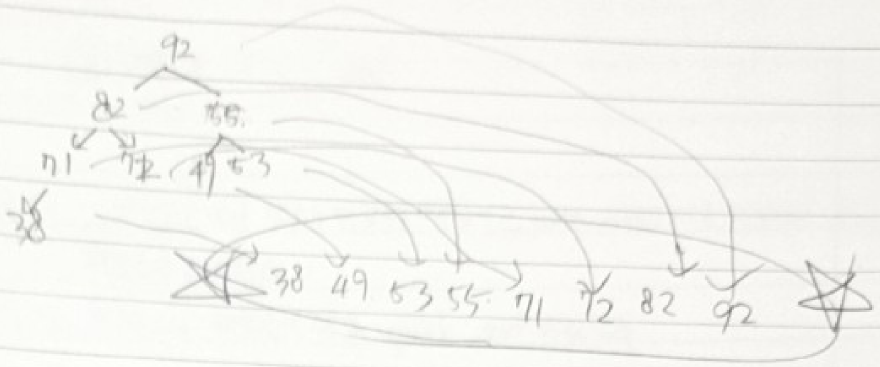
38 49 53 55 71 72 82 92

★ 38 49 53 55 71 72 82 92 ★

(2) 합병정렬



(3)
힙 정렬



07.

(1) ① | 2. 3 4 5 6 7 8

↓

1. 1 ② 3 4 5 6 7 8

↓

1 2 | ③ 4 5 6 7 8

1 2 3 | ④ 5 6 7 8

↓

1 2 3 4 | ⑤ 6 7 8

↓

1 2 3 4 5 ⑥ 7 8

↓

1 2 3 4 5 6 ⑦ 8

↓

1 2 3 4 5 6 7 ⑧

(2). 1 2 3. 4 5 .6 7 8

09. ③ 10. ②

11. ⑤ 7 4 9 8 5 6 3

(a)

☆ 3 4 ⑤ 7 9 8 5 6 ☆

(b) 1번

(c) 변경되지 않는다. 그 피봇값은 이미 정렬이 된것이기 때문이고, 나머지 2개의 서로 배열에서 새로운 피봇들이 선택된다

(d) 왼쪽, 오른쪽 각각의 서브 배열들에 대한 quick_sort 순환 호출.

12.

i)

0	1	2	3	4	5	6	7	8	9
210 220			003 123 513	294				398 528	129 409

~~210~~ ~~220~~ ~~003~~ ~~123~~ ~~513~~ 294 398 ~~528~~ ~~129~~ ~~409~~

ii)

0	1	2	3	4	5	6	7	8	9
003 409	210 513	123 129 220 528		294				398	

~~003~~ ~~409~~ ~~210~~ ~~513~~ ~~123~~ ~~129~~ ~~220~~ ~~528~~ ~~294~~ ~~398~~

iii)

0	1	2	3	4	5	6	7	8	9
003	123 129	210 220 294	398	409	513 528				

☆ 003 -123 -129 - 210 -220 - 294 -398 - 409 - 513 - 528 ☆

13.

```
void insertion_sort (record * rd, int n ){
  int i, j; int tmp;
  for (i=1; i <n; i++){
    tmp = rd[i]->key;
    for (j=i-1; j>=0 && rd[j]->key > tmp; j--)
      rd[j+1] = rd[j];
    rd[j+1]=tmp;
  }
}
```

14.

```
void insertion_sort(). {
  int i, j, key; int tmp = 0;
  int list[6];
  for (int a = 0 ; a < 6; a++){
     scanf("%d", list + a);
  }.

  for ( i=1 ; i < n ; i++) {
     key = list[i];
     for (j = i-1 ; j >= 0 && list[j] > key ; j--)
        list[j+1] = list[j];
     list[j+1] = key; tmp++;
     printf("(");
      for (int b = 0 ; b < tmp ; b++) {
        printf("%d", list[b]);
         if ( b == (tmp-1)) {
            printf(")");
            break;
         }
            printf(",");
       }
      printf("(");
      for (int z = tmp ; z < 6 ;  z++ ) {
        printf("%d", list[z]);
         if ( z == (6 -1)){
           printf(")");
           break;
         }
           printf(",");
        }
     }
}
```

18.

```
void motQsort (int []arr, int front, int rear) {
    int i, j, pivot, mid = front + (rear - front) / 2;
    threesort (arr, front, mid, rear);
    if (rear - front + 1 > 3) {
        pivot = arr[mid];
        swap (arr, mid, rear-1);
        i = front;
        j = rear-1;
        while (true) {
            while (arr[++i] < pivot && i < rear);
            while (arr[--j] > pivot && front < j);
            if (i >= j) break;
            swap (arr, i, j);
        }
        swap (arr, i, rear - 1);
        motQsort (arr, front, i - 1);
        motQsort (arr, i+1, rear);
    }
}
```

☆20.

41 67 34 0 69 24 78 58 62 64

78 69 67 41 62 64 24 34 58 0

21.