자료구조 실습 보고서

[제 10 주]

제출일 : 2018.5.10.목요일

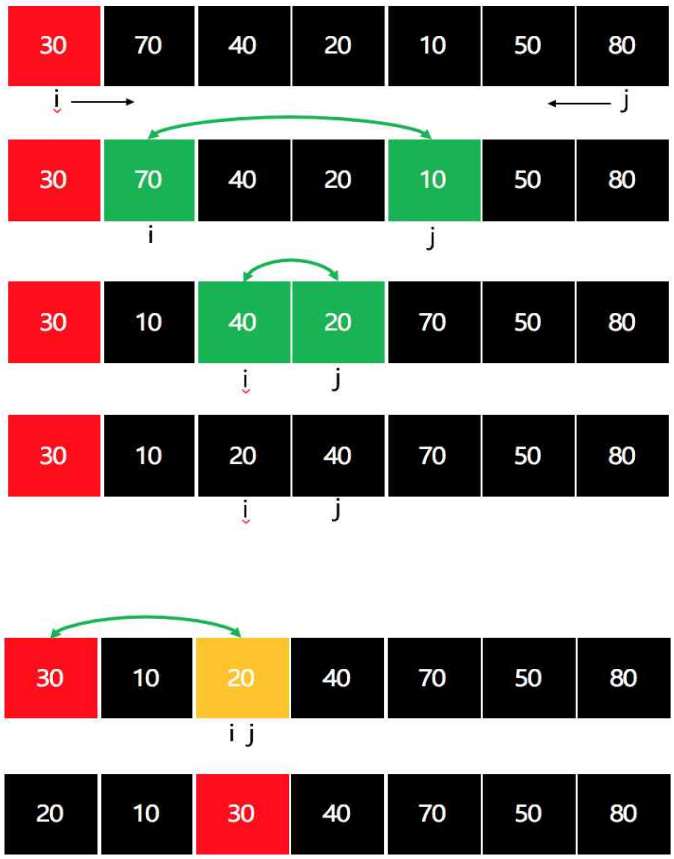
학번 : 201604137

이름 : 김예지

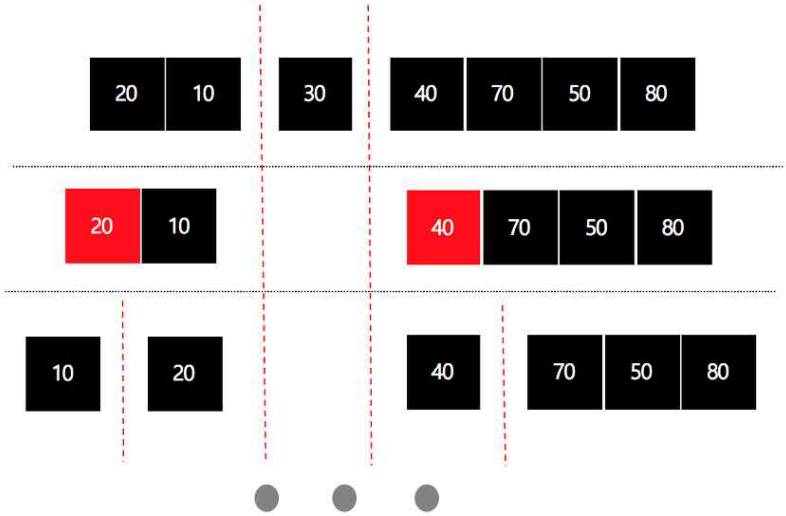
[프로그램 설명서]

퀵 정렬(quick sort)은 기준키(리스트 가운데서 하나의 원소를 고른다. 이렇게 고른 원소를 피벗이라고 한다)를 기준으로 작거나 같은 값을 지닌 데이터는 앞으로, 큰 값을 지닌 데이터는 뒤로 가도록 하여 작은 값을 갖는 데이터와 큰 값을 갖는 데이터로 분리해가며 정렬하는 방법이다.
피벗을 선택하는 방법에는 여러가지가 존재한다. 크게 첫 번째 요소, 중간 요소, 마지막 요소, 랜덤으로 선택하는 방식이 있다.
근데 어떤 피벗을 선택 하든지 기본적으로 작은 값은 왼쪽으로, 큰 값은 오른쪽으로 swap시키기 때문에 크게 상관없다.

1.피벗을 선택한다.
2.오른쪽(j)에서 왼쪽방향으로 가면서 피벗보다 작은 수를 찾는다.
3.왼쪽(i)에서부터 오른쪽으로 가면서 피벗보다 큰 수를 찾는다.
4.각 인덱스 i,j에 대한 요소를 교환한다.
5. 2~4번 과정을 반복한다.
6. 2~3번을 더 이상 진행할 수 없다면, 현재 피벗과 교환한다.
7.그 결과 교환된 피벗을 기준으로 왼쪽은 피벗보다 작은 수들이 존재하고, 오른쪽은 큰 수들이 존재하게된다!




20, 10 <
30
< 40, 70, 50, 80
20
10
30
40
70
50
80


20
10
30
40
70
50
80
20
10
40
70
50
80
10
20
40
70
50
80

[실행 결과 분석]

```
public QuickSort(int size) {
    this.maxSize = size;
    this.array = new int[maxSize];
}

public void sorting() {
    this.quickSortRecursively(0, maxSize - 1);
}

public void quickSortRecursively(int left, int right) {
    if (left < right) {
        int mid = partition(left, right);
        this.quickSortRecursively(left, mid-1);
        this.quickSortRecursively(mid+1, right);
    }
}

public int partition(int left, int right) {
    int pivot = left;
    int data = this.array[pivot];
    right++;
    do {

        while (array[--right] > data) {

        }

        while (array[left] <= data && left < right) {
            left++;
        }

        swap(left, right);

    } while (left < right);

    this.swap(pivot, right);

    return right;
}

public void swap(int a, int b) {
    int temp = array[a];
    array[a] = array[b];
    array[b] = temp;
}
```

메인에서 입력받은 값들을 가지고 sorting()을 호출하고 sorting()에서 quickSortRecursively()를 호출한다. 배열의 처음=left , 마지막=right로 받아와서 partition에 넘겨준다. partition에서는 pivot값이 right보다 작을 때까지 돌린 후에 그 다음 while문으로 들어가서 pivot값이 left보다 크면서 left<right인 지점에서 while문을 나오고 그 두 값을 바꿔준다. 그리고 이 작업이 마무리 되면 pivot과 right를 바꾸고 right를 반환해준다. 그 값이 mid가 되고 이 mid를 가지고 반으로 나눠서 다시 quickSortRecursively()를 호출한다(재귀). 그리고 또 partition을 호출하고 mid를 반환받고를 반복하며 정렬한다.