알고리즘

과제번호	04주차
날 짜	2018.10.04
학 번	201302395
이 름	류경빈

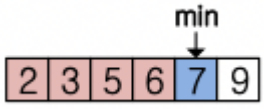
문제 1(2.2-2) 선택 정렬

◆ 선택 정렬 의사 코드

```
1   int min; // 최소 값 저장
2   int temp; // 교체 값 저장
3   selectionSort(Array[], n) {
4           for i in length(array)-1: // array 길이 -1만큼 실행
5                   min = i
6                   for j in (i+1 to length(array)): // i+1=j 값 과 비교
7                           if array[j] 〈 array[min] // 최소 값 과 비교
8                                   min = j
9           temp = array[i]
10          array[i] = array[min]
11          array[min] = temp
```

◆ n개를 바꾸지 않고 n-1개만 바꾸면 되는 이유

SelectionSort를 진행하게 되면 n-1번째 min 값은 여태까지 앞서 정렬 된 값들보다 무조건 큰 값을 가지게 된다. 그럴 경우 n-1번 째 값과 n번째 값을 비교해서 sort 해줬을 경우는 이미 n보다 앞에 있는 배열들이 모두 정렬 되었으며, n번째 값은 다른 배열의 값들보다 무조건 크기 때문이다.



◆ 최선의 경우와 최악의 경우의 수행시간 분석

선택 정렬의 경우 최선의 경우 각 순서마다 정렬되지 않은 범위에서 가장 작은 원소를 찾아 맨 앞자리 값의 자리와 변경한다. 가장 작은 값을 찾아 맨 왼쪽 원소와 변경해 오름차순으로 정렬한다. 의사코드에서 볼 수 있듯이 첫 번째 for 루프는 n-1번 반복되고, 두 번째 for 루프에서는 가장 작은 수를 찾기 위한 비교 상수 시간 작업이기 때문에

T(n) = (n-1) + (n-2) + … + 2 + 1 = n(n-1)/2 = O(n^2)

시간복잡도 : O(n^2)

문제 2(2.3-7) 버블 정렬

◆ 배열에서 두 원소의 합이 x가 되는 경우가 있는지를 알아내는 Θ(nlgn)시간알고리즘 작성

```
1   array A, int x
2   A ← MergeSort(A)
3   for i ←  to length(A) do
4           if A[i] 〉= 0 and BinarySearch(A[i]-x) then
5                   return true
6   end for
7           return false
```

먼저 배열 A를 Merge Sort를 진행 해 오름차순으로 배열을 정렬한다. = O(nlogn)
배열 A에서 Binary Search를 통해 A[i]-x 값을 찾는다. = O(nlogn)
x=A[i]+A[k]일 때, A[k] = -(A[i]-x)이다.

◆ **버블 정렬 의사 코드**

```
1   array A
2   BubbleSort(A[]){
3           for last ← downto 2{
4                   for I ← 1 to last -1
5                           if(A[i]〉A[i+1] then A[i] 〈-〉 A[i+1] // 교환
6           }
7   }
```

◆ **Θ(nlgn)시간복잡도를 가지는 이유 설명**

첫 번째 for 루프는 n-1번 반복이 되고 안쪽 두 번째 for 루프는 n-1, n-2, … , 2, 1번 반복이 진행이 된다. 교환 진행은 상수시간 작업이다.

T(n) = (n-1) + (n-2) + … + 2 + 1 = n(n-1)/2 = O(n^2)

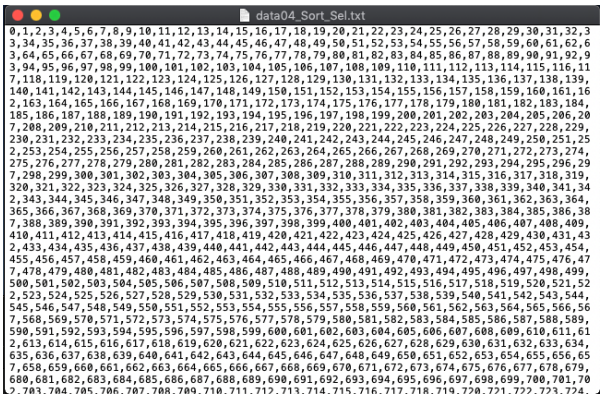
시간복잡도 : O(n^2)

문제 3 선택 정렬 구현

```
public void doSelectionSort(){
    int size = array.size();
    int min;
    int temp;

    for (int i = 0; i < size-1; i++){
        min = i;
        for (int j=i+1; j < size; j++){
            if(array.get(min) > array.get(j)){
                min = j;
            }
        }
        temp = array.get(min);
        array.set(min, array.get(i));
        array.set(i, temp);
    }
}
```

doSelectionSort() 함수 구현

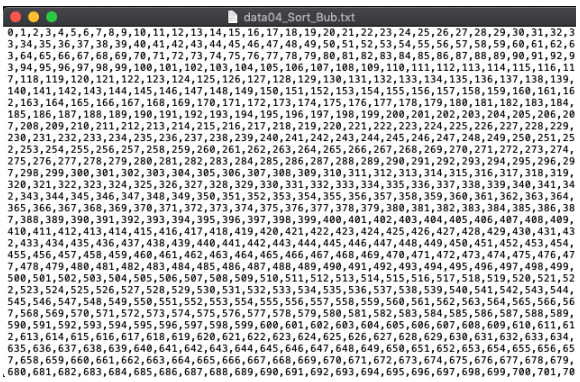


doSelectionSort() 함수를 통해 data04_Sort_Sel.txt 파일 생성

문제 4 버블 정렬 구현

```
public void doBubbleSort(){
    int temp = 0;
    for (int i = array.size()-1; i>=0; i--){
        for (int j=0; j<i; j++){
            if (array.get(j) > array.get(j+1)){
                temp = array.get(j);
                array.set(j, array.get(j+1));
                array.set(j+1, temp);
            }
        }
    }
}
```

doBurbbleSort() 함수 구현



doBubbleSort() 함수를 통해 data04_Sort_Bub.txt 파일 생성