

Nonlinear Data Structure

<http://smartlead.hallym.ac.kr>

Instructor: **Jin Kim**
010-6267-8189(033-248-2318)

jinkim@hallym.ac.kr

Office Hours:
Office : 자연대 7618



Lab(Sorting)

<http://smartlead.hallym.ac.kr>

Instructor: **Jin Kim**
010-6267-8189(033-248-2318)

Office Hours: **jinkim@hallym.ac.kr**



Sorting($O(n^2)$)

- ◆ 삽입정렬 알고리즘을 구현하라(Implement insertionSort algorithm).
- ◆ 선택정렬 알고리즘을 구현하라(Implement selectionSort algorithm).
- ◆ 버블정렬 알고리즘을 구현하라(Implement bubble Sort algorithm).
- ◆ 버블정렬 알고리즘을 수정하여 정렬이 되면 프로그램을 끝내도록 하는 프로그램을 작성하라(Modifiy bubbleSort to FlagbubbleSort algorithm to finish the program if the data is sorted).

Random Number generation 1(난수만들기1)

No need to import

◆ Math.random

a random double from 0.0 (inclusive) to 1.0(exclusive, 즉 0~0.9999999)

```
1 public class MathRandom {  
2  
3     public static void main(String[] args) {  
4         for (int i=0; i<10; ++i)  
5             System.out.println(Math.random());  
6     }  
7 }
```

0~9정수 :

```
for(int i =0; i<100; i++){  
    int randomInt = (int)(10.0 * Math.random());  
}
```

Random Number generation 2(난수만들기2)

(Input data generation)

◆ Java.util.Random.nextInt (random int랜덤 정수)

```
1 import java.util.Random;  
2   
3 public class RandomNextInt {  
4     public static void main(String[] args) {  
5         Random R = new Random();  
6         for (int i=0; i<10; ++i)  
7             System.out.println(R.nextInt(10));  
8     }  
9 }
```

Measuring Elapsed Time(or Execution Time)(실행시간)

```
import java.util.concurrent.TimeUnit;

public class TestElapsedTime {

    static void doSomething() {
        ...
    }

    public static void main(String[] args) {
        long startTime = System.currentTimeMillis();
        doSomething(); //sorting()
        long endTime = System.currentTimeMillis();

        // calculate time difference
        long timeDiff = endTime - startTime;
        System.out.println("Elapsed time in milliseconds: " + timeDiff);
    }
}
```

$\text{timeDiff}/1000 = \text{sec}(\text{초})$

Example(삽입 정렬 실행 시간 계산)

```
import java.util.concurrent.TimeUnit;

public class SortMain {
    private static final int MAX = 100000;
    private static int SIZE = 100000;

    public static void main(String[] args) {
        int a[] = new int[SIZE];
        Random r = new Random();
        for(int i = 0; i < SIZE; i++)
            a[i] = r.nextInt(MAX); //0 <= a[i] < MAX

        long start = System.currentTimeMillis();

        Sorting.insertSort(a);
        long end = System.currentTimeMillis();
        System.out.println("time sort = " + (end - start));
    }
}
```

Insertion sort(n^2)

```
public class Sorting {  
    public static void insertSort(int[] a)  
    {  
        for(int i = 1; i < a.length; i++){  
            int temp = a[i];  
            int j;  
            for(j = i-1; j >= 0 && temp < a[j]; j--)  
                a[j+1] = a[j];  
            a[j+1] = temp;  
        }  
    }  
}
```


SelectionSort(n^2)

```
void selectionsort(int arr[]) {
    int n = arr.length;

    // One by one move boundary of unsorted subarray
    for (int i = 0; i < n-1; i++) {
        // Find the minimum element in unsorted array
        int min_idx = i;
        for (int j = i+1; j < n; j++)
            if (arr[j] < arr[min_idx])
                min_idx = j;
        // Swap the found minimum element with the first
        // element
        int temp = arr[min_idx];
        arr[min_idx] = arr[i];
        arr[i] = temp;
    }
}
```

bubbleSort

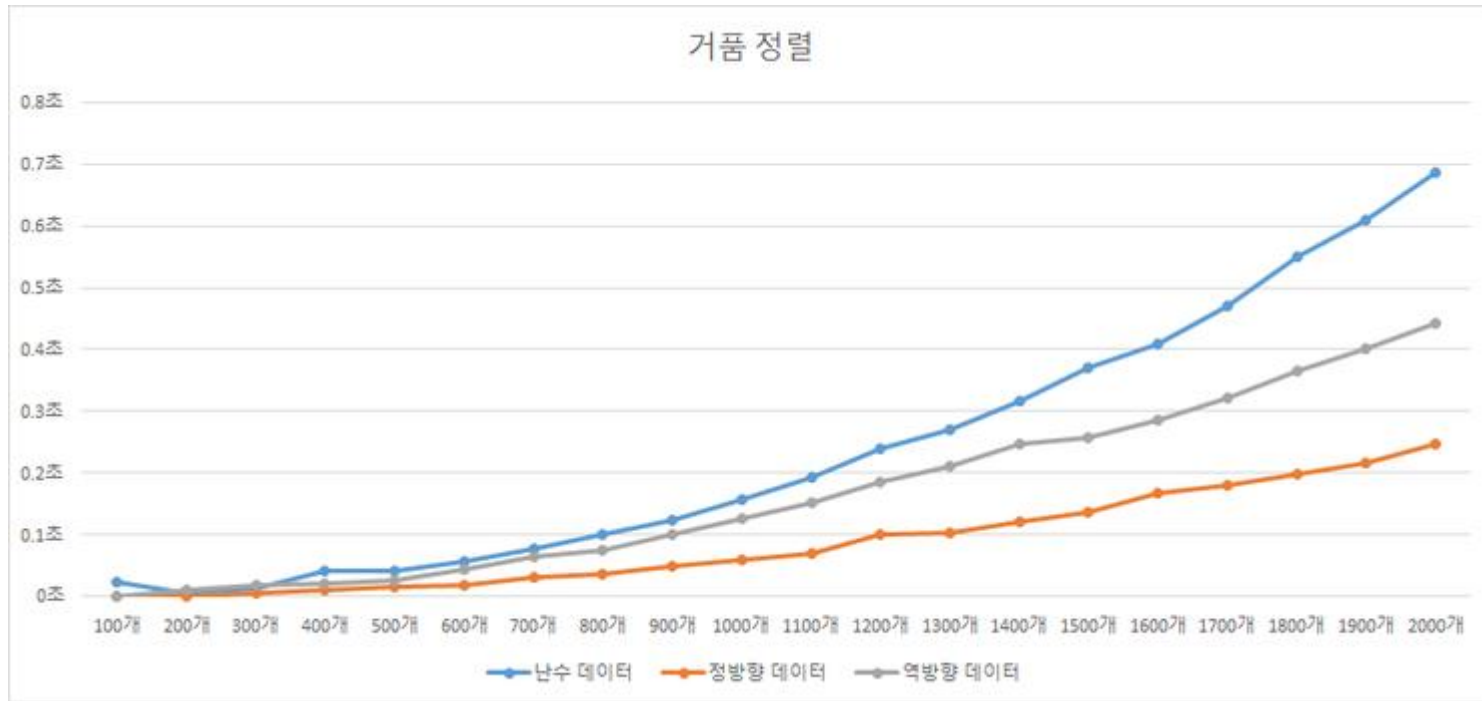
```
static void bubbleSort(int arr[], int n)  {  
    int i, j, temp;  
    boolean swapped;  
    for (i = 0; i < n - 1; i++)    {  
        for (j = 0; j < n - i - 1; j++) {  
            if (arr[j] > arr[j + 1]) {  
                // swap arr[j] and arr[j+1]  
                temp = arr[j];  
                arr[j] = arr[j + 1];  
                arr[j + 1] = temp;  
            }  
        }  
    }  
}
```

FlagbubbleSort(n^2)

```
static void bubbleSort(int arr[], int n)  {
    int i, j, temp;
    boolean swapped;
    for (i = 0; i < n - 1; i++)    {
        swapped = false;
        for (j = 0; j < n - i - 1; j++)    {
            if (arr[j] > arr[j + 1])    {
                // swap arr[j] and arr[j+1]
                temp = arr[j];
                arr[j] = arr[j + 1];
                arr[j + 1] = temp;
                swapped = true;
            }
        }
        // IF no two elements were
        // swapped by inner loop, then break
        if (swapped == false)
            break;
    }
}
```

다음과 같은 그래프를 엑셀(sorting.xlsx)로 만들어라
삽입/선택/버블/flagbubble

Maka graphs(insertion, selection, bubble, flagbubble) using excel.



Input data 1. Random 2. sorted(정방향) 3. reversely sorted(역방향)

네 개의 결과를 하나의 엑셀 파일에 저장하여 업로드하라.

1. Sortmain.java

2. sorting.xlsx(insertion, selection, bubble, flagbubble)

Upload your files at smartlead.hallym.ac.kr





감사합니다.

