

Lab : Sorting(3)

Sorting3

1. ArraysSort방법과 퀵정렬방법비교 : java에서 제공하는 함수를 사용하여 정렬한다. 프로그램은 다음 페이지에 있다. 입력데이터가 random하게 되어 있을때 소요시간을 계산하라. 이때 데이터는 1만,2만,3만,4만,...10만개의 데이터를 사용하라(입력데이터는 0에서 10만미만의 정수). X축은 데이터, y축은 시간으로 하는 2차원표를 엑셀을 사용하여 구현하라. 기존에 작성했던 가장 빠른 정렬 알고리즘인 퀵 정렬과 비교하는 **엑셀표(Fig-1)**을 작성하라. (ArraysSort) 이때 사용하는 데이터는 1)랜덤 2) 정렬 3) 역정렬. 따라서 표는 세개가 나올 것이다.
2. 셸 정렬을 구현하라. 구현한 셸 정렬을 삽입정렬과 비교하는 **그래프(Fig-2)**를 엑셀로 만들어라. (Shell) 이때는 랜덤 데이터만을 이용하여 비교한다.

위의 프로그램들과 엑셀파일을 업로드하라.

자바에서 제공하는 정렬 방법

```
import java.util.Arrays; //자바에서 제공하는 메소드가 저장된 있는 빵틀

public class Arraysort {
    public static void main(String args[]) throws Exception {
        int array[] = { 2, 5, -2, 6, -3, 8, 0, -7, -9, 4 };
        //Arrays.sort(array, 2, 7);
        Arrays.sort(array); // 정렬
        printArray("Sorted array", array);
        int index = Arrays.binarySearch(array, 4); //이진탐색
        System.out.println("Found 4 @ " + index); //4가 몇번방에 있나?
    }
    private static void printArray(String message, int array[]) {
        System.out.println(message + ": [length: " + array.length + "]");
        for (int i = 0; i < array.length; i++) {
            if(i != 0){
                System.out.print(", ");
            }
            System.out.print(array[i]);
        }
        System.out.println();
    }
}
```

셸 정렬 (4)

◆ ShellSort 알고리즘

```
shellSort(a[])  
  interval ← a.length;  
  while (interval > 1) do {  
    interval ← 1 + interval / 3;  
    for (i ← 0; i < interval; i ← i+1) do {  
      intervalSort(a, i, interval);  
    }  
  }  
end ShellSort()
```

셸 정렬 (5)

◆ intervalSort 알고리즘

```
intervalSort(a, i, interval)
// 서브리스트를 삽입 정렬로 정렬하는 ShellSort()의 보조 함수
j ← i + interval;
while (j < a.length) do {
    new ← a[j]; // 서브리스트의 새로운 원소
    k ← j; // new보다 큰 원소는 interval만큼 오른쪽으로 이동
    move ← true;
    while (move) do {
        if (a[k - interval] ≤ new) then {
            move ← false;
        }
        else {
            a[k] ← a[k - interval];
            k ← k - interval;
            if (k = i) then
                move ← false;
        }
    }
    a[k] ← new; // 이동해서 생긴 자리에 삽입
    j ← j + interval; // 다음 서브리스트 원소의 인덱스
}
end intervalSort()
```

셸 정렬 (6)

◆ Sorting 클래스의 메소드 멤버 구현

```
public static void shellSort(int[] a) {  
    int interval = a.length;  
    . . . . . // ShellSort 알고리즘의 Java 코드  
    intervalSort(a, i, interval);  
    . . . . . // 기타 Java 코드  
}  
  
private static void intervalSort(int[] a, int i, int interval) {  
    // 서브리스트를 삽입 정렬로 정렬하는 shellSort()의 보조 메소드  
    . . . . . // intervalSort 알고리즘의 Java 코드  
}
```

1. ArraysSort와 ShellSort를 업로드하라
2. 실행속도와 관련된 엑셀 그래프를 완성하라.

Upload your files at smartlead.hallym.ac.kr





감사합니다.

