

//202201612 이용찬: 작업 스케줄링 문제

```
package Machine;
import java.util.ArrayList;
import java.util.Collections;
import java.util.Comparator;

public class Main {
    public static void main(String[] args) {
        ArrayList<Job> jobs = new ArrayList<Job>();
        jobs.add(new Job(0, 2));
        jobs.add(new Job(1, 6));
        jobs.add(new Job(1, 5));
        jobs.add(new Job(3, 7));
        jobs.add(new Job(5, 9));
        jobs.add(new Job(6, 8));
        jobs.add(new Job(7, 8));
        schedule(jobs);
    }

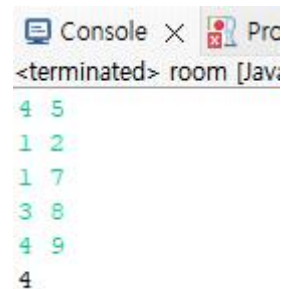
    public static void schedule(ArrayList<Job> jobs) {
        // 일들을 시작 시간을 기준으로 정렬
        Collections.sort(jobs, new Comparator<Job>() {
            public int compare(Job j1, Job j2) {
                return j1.startTime - j2.startTime;
            }
        });

        // 각 기계가 처리하는 일들의 종료 시간을 저장하는 리스트
        ArrayList<Integer> machines = new ArrayList<Integer>();

        for (Job job : jobs) {
            boolean assigned = false;
            // 모든 기계를 확인하여 처리 가능한 기계를 찾음
            for (int i = 0; i < machines.size(); i++) {
                if (machines.get(i) <= job.startTime) {
                    machines.set(i, job.endTime);
                    System.out.println("Job " + job.id + " is assigned to Machine " +
                        (i + 1));
                    assigned = true;
                    break;
                }
            }
            // 처리 가능한 기계가 없으면 새로운 기계를 추가함
            if (!assigned) {
                machines.add(job.endTime);
                System.out.println("Job " + job.id + " is assigned to new Machine " +
                    machines.size());
            }
        }
    }

    class Job {
        int id;
        int startTime;
        int endTime;

        public Job(int startTime, int endTime) {
            this.startTime = startTime;
            this.endTime = endTime;
            this.id = startTime * 10 + endTime;
        }
    }
}
```



```
Console X Prd
<terminated> room [Java
4 5
1 2
1 7
3 8
4 9
4
```

//202201612 이용찬: 백준 1931번 문제

package Machine;

import java.util.\*;

```
public class room {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt(); // 회의의 수
        int[][] times = new int[n][2]; // 회의 시간을 담을 배열

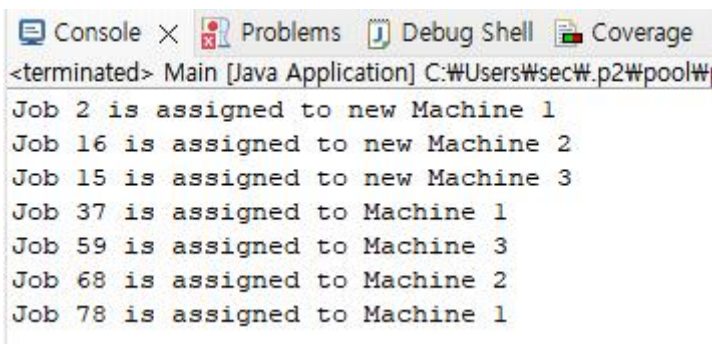
        // 회의 시간 입력 받기
        for (int i = 0; i < n; i++) {
            times[i][0] = sc.nextInt(); // 시작 시간
            times[i][1] = sc.nextInt(); // 끝나는 시간
        }

        // 회의 끝나는 시간 기준 오름차순 정렬
        // 회의 끝나는 시간이 같다면, 시작 시간 기준 오름차순 정렬
        Arrays.sort(times, (a, b) -> {
            if (a[1] == b[1]) {
                return a[0] - b[0];
            } else {
                return a[1] - b[1];
            }
        });

        int count = 1; // 선택한 회의 수 (첫 번째 회의는 무조건 선택)
        int end = times[0][1]; // 이전에 선택한 회의의 끝나는 시간

        // 그리디 알고리즘을 이용하여 최대한 많은 회의 선택
        for (int i = 1; i < n; i++) {
            if (times[i][0] >= end) { // 이전 회의가 끝난 시간 이후에 시작하는 회의인
                count++; // 회의 선택 수 증가
                end = times[i][1]; // 선택한 회의의 끝나는 시간으로 갱신
            }
        }

        System.out.println(count); // 선택한 회의 수 출력
    }
}
```



```
Console × Problems Debug Shell Coverage
<terminated> Main [Java Application] C:\Users\sec#.p2\pool\#
Job 2 is assigned to new Machine 1
Job 16 is assigned to new Machine 2
Job 15 is assigned to new Machine 3
Job 37 is assigned to Machine 1
Job 59 is assigned to Machine 3
Job 68 is assigned to Machine 2
Job 78 is assigned to Machine 1
```