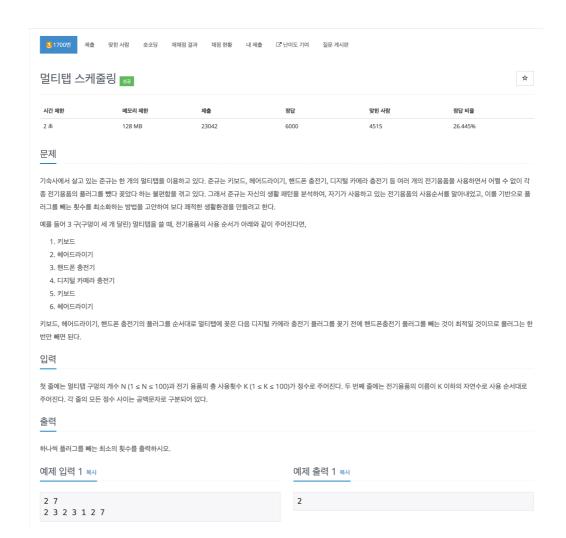
BOJ 1700: 멀티탭 스케줄링



문제





문제 풀이과정 정답

풀이과정

아직은 낯선 그리디 문제.

처음에는, 카운트배열을 생성하여 가장 많이 사용하는 순으로 문자를 정렬한 다음 초기 콘센트에 가장 많이 사용하는 것들을 꽂아두고 시작하는 식으로 하였으나 실패. (가장 많이 사용하는 코드를 뽑았다가 꽂았다가를 반복하기에 절대 최소가 나올 수 없다.)

문제를 해결한 최적해는 다음과 같다.

먼저 콘센트가 비어있다면, 빈 곳에 콘센트를 꽂는다.

콘센트가 꽂아져 있다면 continue;

어떠한 콘센트를 뽑아야하는 상황이라면, 더이상 사용하지 않거나 가장 나중에 사용되는 콘센트를 뽑는다.

정답

```
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.util.*;
public class Main {
   public static void main(String[] args) throws Exception {
       BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
       StringTokenizer st = new StringTokenizer(br.readLine());
        int N = Integer.parseInt(st.nextToken());
       int K = Integer.parseInt(st.nextToken());
       int[] arr = new int[K];
       st = new StringTokenizer(br.readLine());
        for (int i = 0; i < K; i++) {
           arr[i] = Integer.parseInt(st.nextToken());
       List<Integer> list = new ArrayList<>();
        for(int i=0; i<N; i++) {</pre>
           list.add(0);
       int cnt = 0;
        for (int i = 0; i < K; i++) {
           if (list.indexOf(arr[i]) >= 0) continue;
            if (list.indexOf(0) >= 0) {
               list.set(list.indexOf(0),arr[i]);
                continue:
            }
           int maxCnt = -1;
            int currIdx = -1;
            for (int k = 0; k < N; k++) {
                int temp = 0;
                for (int j = i+1; j < K; j++) {
                   if(arr[j] == list.get(k)) break;
                   temp++;
                if(temp >= maxCnt) {
                   maxCnt = temp;
                   currIdx = k;
            }
           list.set(currIdx, arr[i]);
            cnt++;
        }
       System.out.println(cnt);
   }
}
```