

1. "한 번 사용한 최초문자" 정답코드

```
class Solution {  
    public int solution(String s){  
        HashMap<Character, Integer> sH = new HashMap<>();  
        for(char x : s.toCharArray()){  
            sH.put(x, sH.getOrDefault(x, 0) + 1);  
        }  
        for(int i = 0; i < s.length(); i++){  
            if(sH.get(s.charAt(i)) == 1) return i+1;  
        }  
        return -1;  
    }  
}
```

▶▶ Comment :

문자열을 해싱한 후 빈도수가 1인 문자 중 문자열에서 제일 앞에 위치한 문자를 찾으면 됩니다.

특히 문자열을 앞에서부터 탐색하면서 각 문자의 빈도수가 1인가 확인해 빈도수가 1인 문자가 최소로 발견되면 이 문자의 인덱스가 답이 됩니다.

2. "같은 빈도수 만들기" 정답코드

```

class Solution {
    public int[] solution(String s){
        int[] answer = new int[5];
        HashMap<Character, Integer> sH = new HashMap<>();
        for(char x : s.toCharArray()){
            sH.put(x, sH.getOrDefault(x, 0)+1);
        }
        int max = Integer.MIN_VALUE;
        String tmp = "abcde";
        for(char key : tmp.toCharArray()){
            if(sH.getOrDefault(key, 0) > max){
                max = sH.getOrDefault(key, 0);
            }
        }
        for(int i = 0; i < tmp.length(); i++){
            answer[i] = max - sH.getOrDefault(tmp.charAt(i), 0);
        }
        return answer;
    }
}

```

▶▶ Comment :

```

for(char key : tmp.toCharArray()){
    if(sH.getOrDefault(key, 0) > max){
        max = sH.getOrDefault(key, 0);
    }
}

```

tmp문자열의 문자인 key가 sH 해쉬의 키로 존재하지 않을 수 있습니다.

sH.get(key)로 하면 값이 null값이 될 수 있으니 반드시 sH.getOrDefault(key, 0)로 해야만 합니다.

3. "서로 다른 빈도수 만들기" 정답코드

```

class Solution {
    public int solution(String s){
        int answer = 0;
        HashMap<Character, Integer> sH = new HashMap<>();
        HashSet<Integer> ch = new HashSet<>();
        for(char x : s.toCharArray()){
            sH.put(x, sH.getOrDefault(x, 0) + 1);
        }
        for(char key : sH.keySet()){
            while(ch.contains(sH.get(key))){
                answer++;
                sH.put(key, sH.get(key) - 1);
            }
            if(sH.get(key) == 0) continue;
            ch.add(sH.get(key));
        }
        return answer;
    }
}

```

▶▶ Comment :

```

for(char key : sH.keySet()){
    while(ch.contains(sH.get(key))){
        answer++;
        sH.put(key, sH.get(key) - 1);
    }
    if(sH.get(key) == 0) continue;
    ch.add(sH.get(key));
}

```

char key의 빈도수 sH.get(key)값이 0이면 ch에 빈도수인 0을 추가하면 안됩니다.

while문 안에서 하나씩 삭제되어 빈도수가 0이 되면 "key"라는 문자는 문자열에서 완전히 삭제되었다는 것을 의미합니다. 즉, "key"라는 문자는 존재하지 않고 빈도수 또한 존재하는 빈도수가 아니기 때문에 ch에 추가하지 않습니다.

4. "음수가 있는 부분수열" 정답코드

```
class Solution {  
    public int solution(int[] nums, int m){  
        int answer = 0;  
        HashMap<Integer, Integer> nH = new HashMap<>();  
        int sum = 0;  
        nH.put(0, 1);  
        for(int x : nums){  
            sum += x;  
            if(nH.containsKey(sum-m)) answer += nH.get(sum-m);  
            nH.put(sum, nH.getOrDefault(sum, 0) + 1);  
        }  
        return answer;  
    }  
}
```

5. "회장 선거" 정답코드

```

class Solution {
    public String solution(String[] votes, int k){
        String answer = " ";
        HashMap<String, HashSet<String>> voteHash = new HashMap<>();
        HashMap<String, Integer> candidate = new HashMap<>();
        HashMap<String, Integer> present = new HashMap<>();
        for(String x : votes){
            String a = x.split(" ")[0];
            String b = x.split(" ")[1];
            voteHash.putIfAbsent(a, new HashSet<String>());
            voteHash.get(a).add(b);
            candidate.put(b, candidate.getOrDefault(b, 0) + 1);
        }
        int max=Integer.MIN_VALUE;
        for(String a : voteHash.keySet()){
            int cnt = 0;
            for(String b : voteHash.get(a)){
                if(candidate.get(b) >= k) cnt++;
            }
            present.put(a, cnt);
            max = Math.max(max, cnt);
        }
        ArrayList<String> tmp = new ArrayList<>();
        for(String name : present.keySet()){
            if(present.get(name) == max) tmp.add(name);
        }
        tmp.sort((a, b) -> a.compareTo(b));
        answer = tmp.get(0);
        return answer;
    }
}

```

▶▶ Comment :

```
HashMap<String, HashSet<String>> voteHash = new HashMap<>();
```

해쉬맵의 키의 값으로 자료구조가 사용될 수 있습니다. 위 코드는 해쉬셋을 키의 값으로 사용한 예입니다.

```
voteHash.putIfAbsent(a, new HashSet<String>());
```

putIfAbsent 함수는 a라는 키가 존재하지 않으면 작동합니다. 위 코드는 a키를 생성하고 그 값으로 HashSet 자료구조를 생성합니다.

6. "문서 도난" 정답코드

```

class Info implements Comparable<Info>{
    public String name;
    public int time;
    Info(String name, int time){
        this.name = name;
        this.time = time;
    }
    @Override
    public int compareTo(Info ob){
        return this.time - ob.time;
    }
}

class Solution {
    public int getTime(String time){
        int H = Integer.parseInt(time.split(":")[0]);
        int M = Integer.parseInt(time.split(":")[1]);
        return H*60+M;
    }

    public String[] solution(String[] reports, String times){
        ArrayList<Info> tmp = new ArrayList<>();
        for(String x : reports){
            String a = x.split(" ")[0];
            String b = x.split(" ")[1];
            tmp.add(new Info(a, getTime(b)));
        }
        Collections.sort(tmp);
        int s = getTime(times.split(" ")[0]);
        int e = getTime(times.split(" ")[1]);
        ArrayList<String> res = new ArrayList<>();
        for(Info ob : tmp){
            if(ob.time >= s && ob.time <= e){
                res.add(ob.name);
            }
            if(ob.time > e) break;
        }
        String[] answer = new String[res.size()];
        for(int i = 0; i < res.size(); i++){
            answer[i] = res.get(i);
        }
        return answer;
    }
}

```

7. "경고 메일" 정답코드

```

class Solution {
    public int getTime(String time){
        int H = Integer.parseInt(time.split(":")[0]);
        int M = Integer.parseInt(time.split(":")[1]);
        return H*60+M;
    }
    public String[] solution(String[] reports, int time){
        HashMap<String, Integer> inT = new HashMap<>();
        HashMap<String, Integer> sumT = new HashMap<>();
        for(String x : reports){
            String a = x.split(" ")[0];
            String b = x.split(" ")[1];
            String c = x.split(" ")[2];
            if(c.equals("in")) inT.put(a, getTime(b));
            else sumT.put(a, sumT.getOrDefault(a, 0) + (getTime(b) - inT.get(a)));
        }
        ArrayList<String> res = new ArrayList<>();
        for(String name : sumT.keySet()){
            if(sumT.get(name) > time) res.add(name);
        }
        res.sort((a, b) -> a.compareTo(b));
        String[] answer = new String[res.size()];
        for(int i = 0; i < res.size(); i++){
            answer[i] = res.get(i);
        }
        return answer;
    }
}

```

▶▶ Comment :

```

if(c.equals("in")) inT.put(a, getTime(b));
else sumT.put(a, sumT.getOrDefault(a, 0) + (getTime(b) - inT.get(a)));

```

1. 들어오는 시간을 inT에 분으로 환산해서 기록합니다.
2. getTime(b)으로 나가는 시간을 분으로 환산한 시간에서 a가 들어온 시간inT.get(a)을 빼주면 a가 보안실을 이용한 시간입니다. 이용한 시간을 sumT에 누적합니다.