AS_4_오윤기_개인결과물

<5356 의석이의 세로로 말해요>

: 문제의 지문과 반대로 들어와야한다는 것을 체크 못하고 잠깐 헤맸음. for문을 사용해 String배열을 이용하여 탐색한 후 출력함.

SW Expert Academy

SW 프로그래밍 역량 강화에 도움이 되는 다양한 학습 컨텐츠를 확인하세요!

https://swexpertacademy.com/main/code/problem/problemDetail.do?contestProbId=AWVWgkP6sQ0DFAUO&categoryId=AWVWgkP6sQ0DFAUO&categoryType=CODE&problemTitle=%EC%9D%98%EC%84%9D&orderBy=FIRST_REG_DATETIME&selectCodeLang=ALL&select-1=&pageSize=10&pageIndex=1


```
public class Solution {
   public static void main(String[] args) throws FileNotFoundException {
       Scanner sc = new Scanner(System.in);
       int T = sc.nextInt();
       sc.nextLine();
       for (int tc = 1; tc <= T; tc++) {
           String[] arr = new String[5];
           int max = -1;
           for (int i = 0; i < 5; i++) {
               arr[i] = sc.nextLine();
               if (max < arr[i].length())
                   max = arr[i].length();
           }
           System.out.print("#" + tc + " ");
           for (int i = 0; i < max; i++) {
               for (int j = 0; j < 5; j++) {
                   if (arr[j].length() - 1 < i) {
                       System.out.print("");
                       continue;
                   }
                   System.out.print(arr[j].charAt(i));
               }
           }
           System.out.println();
       }
```

```
        sc.close();
    }

}
```

<8320 직사각형을 만드는 방법>

8320번: 직사각형을 만드는 방법

두 직사각형 A와 B가 있을 때, A를 이동, 회전시켜서 B를 만들 수 없으면, 두 직사각형은 다르다고 한다. 직사각형을 만들 때, 정사각형을 변형시키거나, 한 정사각형 위에 다른 정사각형을 놓을 수 없다. 또, 직사각형은 정

https://www.acmicpc.net/problem/8320

가로 세로가 넓의의 인수임을 이용하고

이동또는 회전할 경우 중복되는 사각형임을 고려하여

1~n까지의 i넓이를 가진 사각형을 선택한 후, sqrt(i)까지의 i 약수를 탐색하는 방법으로 구현

```
public class Main8320 {
  public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    int n = sc.nextInt();
    int cnt = 0;

    for (int i = 1; i <=n; i++) {
      for (double j = 1; j <=Math.sqrt(i); j++) {
        if ((i % j) == 0) {
          cnt++;
        }
      }
    }
    System.out.println(cnt);
    sc.close();
  }
}
```

***시도했지만 못푼문제**

<4796 의석이의우뚝선산>

SW Expert Academy

SW 프로그래밍 역량 강화에 도움이 되는 다양한 학습 컨텐츠를 확인하세요!

https://swexpertacademy.com/main/code/problem/problemDetail.do?contestProbId=AWS2h6AKBCoDFAVT&categoryId=AWS2h6AKBCoDFAVT&categoryType=CODE&problemTitle=%EC%9D%98%EC%84%9D&orderBy=FIRST_REG_DATETIME&selectCodeLang=ALL&select-1=&pageSize=10&pageIndex=1

for문을 3번 사용하면 최악의 케이스일 경우 시간초과가 무조건 난다. 어떻게 해결해야할지 고민하던 찰나 같은 팀원의 해결방법을 통하여 최대값을 순차적으로 찾은 후 주변의 영역을 부분집합으로 체크하여 구현하는 방법을 듣고 구현방법이 떠올랐지만 구현시도는 못해봄.

```
public class Solution4796_의석이 {
  static class Mountain implements Comparable<Mountain> {
    int index;
    int height;

    Mountain(int index, int height) {
      this.index = index;
      this.height = height;
    }

    @Override
    public int compareTo(Mountain o) {
      return o.height - this.height;
    }
  }

  static int N;
  static int[] h;
  static Mountain[] mountains;

  public static void main(String[] args) throws FileNotFoundException {
    System.setIn(new FileInputStream("sample_input (2).txt"));
    Scanner sc = new Scanner(System.in);
    int TC = sc.nextInt();
    for (int tc = 1; tc <= TC; tc++) {
      N = sc.nextInt();
      h = new int[N];
      mountains = new Mountain[N];
      for (int i = 0; i < N; i++) {
        h[i] = sc.nextInt();
        mountains[i] = new Mountain(i, h[i]);
      }

      Arrays.sort(mountains);
```

```
      int cnt = 0;
      for (int i = 0; i < N; i++) {
        int peakidx = mountains[i].index;
        int prev = peakidx - 1;
        int next = peakidx + 1;
        if (prev > -1 && next < N && h[prev] < h[peakidx] && h[peakidx] > h[next]) {
          cnt++;
          for (int j = prev; j > 0; j--) {
            if (h[j - 1] < h[j])
              cnt++;
            else {
              for (int k = next; k < N - 1; k++) {
                if (h[k] > h[k + 1])
                  cnt++;
                else
                  break;
              }
            }
          }
        }
      }
      System.out.println("#" + tc + " " + cnt);
    }
  }
}
```