

Day 5: Problem Analysis

Niyaz Nigmatullin

September 12, 2015

Problem A

Problem B

Problem C

Problem D

Problem E

Problem F

Problem G

Problem H

Problem I

Problem J

Problem A. Arrangement of Contest

Day 5: Problem
Analysis

Niyaz
Nigmatullin

```
boolean[] have = new boolean[26];  
for (int i = 0; i < n; i++) {  
    have[name[i].charAt(0) - 'A'] = true;  
}  
int answer = 0;  
while (answer < 26 && have[answer])  
    ++answer;  
out.println(answer);
```

Problem A

Problem B

Problem C

Problem D

Problem E

Problem F

Problem G

Problem H

Problem I

Problem J

Problem B. Auxiliary Question of the Universe

Solution

- ▶ digit $d \rightarrow d^+$
- ▶ $(\text{ or }) \text{ or } + \rightarrow (0)^+$

```
for (char c : in.nextLine().toCharArray()) {  
    if (Character.isDigit(c)) {  
        out.print(c + "+");  
    } else {  
        out.print("(0)+");  
    }  
}  
out.println("0");
```

Problem A

Problem B

Problem C

Problem D

Problem E

Problem F

Problem G

Problem H

Problem I

Problem J

Problem C. Database

Day 5: Problem
Analysis

Niyaz
Nigmatullin

```
for (int i = 0; i < n; i++) {
    f[i] = line[i].split(",");
}
for (int i = 0; i < m; i++) {
    for (int j = i + 1; j < m; j++) {
        Map<String, Integer> where;
        where = new HashMap<>();
        for (int k = 0; k < n; k++) {
            String cur = f[k][i] + "," + f[k][j];
            Integer p = where.get(cur);
            if (p != null) {
                out.println("NO");
                out.println((p + 1) + " " + (k + 1));
                out.println((i + 1) + " " + (j + 1));
                return;
            }
            where.put(cur, k);
        }
    }
}
out.println("YES");
```

Problem A

Problem B

Problem C

Problem D

Problem E

Problem F

Problem G

Problem H

Problem I

Problem J

Problem D. Defense of a Kingdom

Day 5: Problem
Analysis

Niyaz
Nigmatullin

```
if (n == 0) {  
    out.println(w * h);  
    return;  
}  
Arrays.sort(x);  
Arrays.sort(y);  
int mx = Math.max(x[0] - 1, w - x[n - 1]);  
int my = Math.max(y[0] - 1, h - y[n - 1]);  
for (int i = 0; i + 1 < n; i++) {  
    mx = Math.max(mx, x[i + 1] - x[i]);  
    my = Math.max(my, y[i + 1] - y[i]);  
}  
out.println(mx * my);
```

Problem A

Problem B

Problem C

Problem D

Problem E

Problem F

Problem G

Problem H

Problem I

Problem J

Problem E. Explicit Formula

```
int ans = 0;
for (int i = 0; i < n; i++) {
    for (int j = i + 1; j < n; j++) {
        ans ^= a[i] | a[j];
        for (int k = j + 1; k < n; k++) {
            ans ^= a[i] | a[j] | a[k];
        }
    }
}
out.println(ans);
```

Problem A

Problem B

Problem C

Problem D

Problem E

Problem F

Problem G

Problem H

Problem I

Problem J

Problem F. Flat

```
for (int i = 0; i < n; i++) {
    if (t[i].equals("bedroom")) {
        areabed += a[i];
    }
    total += a[i];
    if (t[i].equals("balcony")) {
        cost += 0.5 * c * a[i];
    } else {
        cost += c * a[i];
    }
}
out.println(total);
out.println(areabed);
out.println(cost);
```

Problem A

Problem B

Problem C

Problem D

Problem E

Problem F

Problem G

Problem H

Problem I

Problem J

Problem G. Garage

Day 5: Problem
Analysis

Niyaz
Nigmatullin

```
int mw = (W - w) / (2 * w) + 1;  
int mh = (H - h) / (2 * h) + 1;  
out.println(mw * mh);
```

Problem A

Problem B

Problem C

Problem D

Problem E

Problem F

Problem G

Problem H

Problem I

Problem J

Problem H. Homo or Hetero?

```
read(s, x);
if (s.equals("insert")) {
    a[x]++;
    if (a[x] == 1) hetero++;
    if (a[x] == 2) homo++;
} else {
    a[x]--;
    if (a[x] == 0) hetero--;
    if (a[x] == 1) homo--;
}
if (homo > 0 && hetero > 1) {
    out.println("both");
} else if (homo > 0) {
    out.println("homo");
} else if (hetero > 1) {
    out.println("hetero");
} else {
    out.println("neither");
}
```

Problem A

Problem B

Problem C

Problem D

Problem E

Problem F

Problem G

Problem H

Problem I

Problem J

Problem I. Java vs C++

Solution

- ▶ Underscore and uppercase → error
- ▶ Starts with underscore or uppercase → error
- ▶ All lowercase → output the same
- ▶ Underscore → C++
- ▶ Otherwise → Java
- ▶ Parse and output in another language

Problem A

Problem B

Problem C

Problem D

Problem E

Problem F

Problem G

Problem H

Problem I

Problem J

Problem J. John's Inversions

Solution

- ▶ Suppose it's not sorted by red
- ▶ Let's take to neighboring unsorted elements
- ▶ Swapping them gives -1 to red inversions
 - ▶ But gives no more that +1 to blue ones
- ▶ It's not bad to sort by one of the colors
- ▶ Sort by one of the colors
- ▶ Count number of inversions of another color
- ▶ Time complexity: $O(n \log n)$

Problem A

Problem B

Problem C

Problem D

Problem E

Problem F

Problem G

Problem H

Problem I

Problem J