

Разбор задач олимпиады «Муниципальный этап Всероссийской олимпиады школьников Красноярского края по информатике, 7-8 классы» сайта астр.ru от 07.12.2013 года

Задача А. Точки и прямоугольники

Внутри первого прямоугольника находится $(W-1) \cdot (H-1)$ точек с целыми координатами. А второй прямоугольник содержит ровно $(w+1) \cdot (h+1)$ таких точек. Значит результатом будет разность этих произведений.

В этой задаче важно обратить внимание на ограничения. Все числа могут быть до миллиарда, а это значит, что произведение двух из них будет порядка 10^{18} . Поэтому следует использовать 64-битный тип данных (*long long* или *Int64*).

Решение на языке C++

```
1 #include <iostream>
2 #include <cstdio>
3
4 using namespace std;
5
6 typedef long long LL;
7
8 int main(){
9     freopen("input.txt", "r", stdin);
10    freopen("output.txt", "w", stdout);
11
12    LL W, H, w, h;
13    cin>>W>>H>>w>>h;
14    cout<<(W-w-2)*(H-1)+(w+1)*(H-h-2);
15    return 0;
16 }
```

Решение на языке Паскаль

```
1 var
2   W, H, _w, _h : Int64;
3 begin
4   assign(input, 'input.txt'); reset(input);
5   assign(output, 'output.txt'); rewrite(output);
6
7   Read(W, H, _w, _h);
8   Write((W-_w-2)*(H-1) + (_w+1)*(H-_h-2));
9 end.
```

Задача В. Светофор

Так как количество состояний светофора невелико, то можно просто все их перебрать с помощью составных условий и для каждого варианта вывести правильный ответ.

Предполагаем, что приведённый ниже код не требует пояснений.

Решение на языке C++

```
1 #include <iostream>
```

```
2 #include <stdio>
3 #include <cstring>
4
5 using namespace std;
6
7 int main(){
8     freopen("input.txt", "r", stdin);
9     freopen("output.txt", "w", stdout);
10
11     string s1, s2, s3;
12     cin>>s1>>s2>>s3;
13     if(s1=="black" && s2=="black" && s3=="green"){
14         cout<<s1<<endl<<s2<<endl<<"GREEN";
15         return 0;
16     }
17     if(s1=="black" && s2=="black" && s3=="GREEN"){
18         cout<<s1<<endl<<"yellow"<<endl<<s1;
19         return 0;
20     }
21     if(s1=="black" && s2=="yellow" && s3=="black"){
22         cout<<"red\n"<<s1<<endl<<s1;
23         return 0;
24     }
25     if(s1=="red" && s2=="black" && s3=="black"){
26         cout<<s1<<endl<<"yellow"<<endl<<s3;
27         return 0;
28     }
29     if(s1=="red" && s2=="yellow" && s3=="black"){
30         cout<<s3<<endl<<s3<<endl<<"green";
31         return 0;
32     }
33     if(s1=="black" && s2=="YELLOW" && s3=="black"){
34         cout<<s1<<endl<<s2<<endl<<s3;
35         return 0;
36     }
37     cout<<"error";
38     return 0;
39 }
```

Решение на языке Паскаль

```
1 var
2     s1, s2, s3 : String;
3 begin
4     assign(input, 'input.txt'); reset(input);
5     assign(output, 'output.txt'); rewrite(output);
6
7     ReadLn(s1); ReadLn(s2); ReadLn(s3);
8
9     if (s1='black') and (s2='black') and (s3='green') then begin
10         WriteLn(s1); WriteLn(s2); WriteLn('GREEN');
11     end else
12     if (s1='black') and (s2='black') and (s3='GREEN') then begin
13         WriteLn(s1); WriteLn('yellow'); WriteLn(s1);
14     end else
15     if (s1='black') and (s2='yellow') and (s3='black') then begin
16         WriteLn('red'); WriteLn(s1); WriteLn(s1);
17     end else
18     if (s1='red') and (s2='black') and (s3='black') then begin
19         WriteLn(s1); WriteLn('yellow'); WriteLn(s3);
20     end else
21     if (s1='red') and (s2='yellow') and (s3='black') then begin
22         WriteLn(s3); WriteLn(s3); WriteLn('green');
23     end else
24     if (s1='black') and (s2='YELLOW') and (s3='black') then begin
25         WriteLn(s1); WriteLn(s2); WriteLn(s3);
26     end else WriteLn('error');
27 end.
```

Задача С. Пирожки

Так как нашему герою надо максимизировать количество купленных пирожков, то логично начинать покупки с самого дешёвого из них. Поэтому сначала поменяем местами введённые данные так, чтобы стоимости не убывали.

Тогда пирожков первого типа Петя может купить $\left\lfloor \frac{s}{c_1} \right\rfloor$ (где $\lfloor a \rfloor$ – целая часть от a),

но не более, чем их есть в продаже (то есть не более n_1). Готовая формула получается

$$\min \left\{ n_1, \left\lfloor \frac{s}{c_1} \right\rfloor \right\}.$$

Аналогично рассчитываются покупки пирожков других типов (с учётом того, что часть денег уже потрачена).

Подводным камнем в этой задаче является наличие нулей во входных данных (даже в цене пирожка). Поэтому каждую операцию деления следует обработать в условии.

Решение на языке C++

```
1 #include <iostream>
2 #include <cstdio>
3 #include <algorithm>
4
5 #define MIN(a,b) ((a)<(b)?(a):(b))
6
7 using namespace std;
8
9 typedef long long LL;
10
11 int main(){
12     freopen("input.txt", "r", stdin);
13     freopen("output.txt", "w", stdout);
14
15     LL c1, c2, c3, n1, n2, n3, s;
16     cin>>c1>>c2>>c3>>n1>>n2>>n3>>s;
17     if(c1>c2) swap(c1,c2), swap(n1,n2);
18     if(c2>c3) swap(c2,c3), swap(n2,n3);
19     if(c1>c2) swap(c1,c2), swap(n1,n2);
20
21     LL r=0, t;
22     r = MIN(n1, (c1?s/c1:n1));
23     s-=r*c1;
24     t = MIN(n2, (c2?s/c2:n2));
25     s-= t*c2;
26     r += t;
27     r += MIN(n3, (c3?s/c3:n3));
28     cout<<r;
29
30     return 0;
31 }
```

Решение на языке Паскаль

```
1 uses math;
2 var
3     c1, c2, c3, n1, n2, n3, s, t, r, tmp : Int64;
4 begin
5     assign(input, 'input.txt'); reset(input);
6     assign(output, 'output.txt'); rewrite(output);
7
8     Read(c1, c2, c3, n1, n2, n3, s);
9
10    if c1 > c2 then begin
11        t := c1; c1 := c2; c2 := t;
12        t := n1; n1 := n2; n2 := t;
```

```
13 end;
14 if c2 > c3 then begin
15     t := c2; c2 := c3; c3 := t;
16     t := n2; n2 := n3; n3 := t;
17 end;
18 if c1 > c2 then begin
19     t := c1; c1 := c2; c2 := t;
20     t := n1; n1 := n2; n2 := t;
21 end;
22
23 if c1 <> 0 then tmp := s div c1 else tmp := n1;
24 r := min(n1, tmp);
25 dec(s, r*c1);
26 if c2 <> 0 then tmp := s div c2 else tmp := n2;
27 t := min(n2, tmp);
28 dec(s, t*c2);
29 inc(r, t);
30 if c3 <> 0 then tmp := s div c3 else tmp := n3;
31 inc(r, min(n3, tmp));
32 Write(r);
33 end.
```

Задача D. Сжатие бинарных последовательностей

В этой задаче надо понимать, что все буквы тоже имеют свой числовой код (ASCII-код). Например буква «а» имеет код 97. А также к ним тоже можно прибавлять целые числа, получая другие символы. Например, если к «а» прибавить один, то получим «б». В С++ это можно так и писать, а в Паскале всё приходится делать с помощью функций *ord* и *chr*.

Одним проходом по строке будем считать количество подряд идущих нулей, а когда встречаем единицу, то сразу выводить полученную букву.

Решение на языке С++

```
1 #include <iostream>
2 #include <cstdio>
3
4 using namespace std;
5
6 int main(){
7     freopen("input.txt", "r", stdin);
8     freopen("output.txt", "w", stdout);
9
10    char s[999];
11    cin>>s;
12    for(int i=0, c=0;s[i];++i)
13        if(s[i]=='1'){
14            cout<<(char)('a'+c);
15            c=0;
16        }else ++c;
17    return 0;
18 }
```

Решение на языке Паскаль

```
1 var
2     s : AnsiString;
3     i, c : Longint;
4 begin
5     assign(input, 'input.txt'); reset(input);
6     assign(output, 'output.txt'); rewrite(output);
7
8     ReadLn(s);
9     for i:=1 to length(s) do
10        if s[i] = '1' then begin
11            Write(chr(97+c));
```

```
12         c := 0;
13     end else inc(c);
14 end.
```

Задача Е. Бег по эскалатору

Так как на эскалаторе нельзя обгонять, то расположим всех людей в порядке увеличения стартовой ступеньки. Это можно сделать с помощью алгоритма сортировки. Количество людей достаточно велико, поэтому квадратичные алгоритмы не подходят для этой задачи. В C++ можно воспользоваться стандартной библиотекой шаблонов и применить процедуру `sort`. В Паскале же придётся написать быструю сортировку Хоара (или любой другой алгоритм сортировки с асимптотикой времени работы не хуже $O(N \log N)$) самостоятельно.

Заметим, что нет необходимости сортировать сами данные. Можно завести ещё один массив с номерами людей и сортировать его. То есть расположить номера людей в порядке возрастания стартовой ступеньки (такой вариант реализован в решении на Паскале).

Если бы на эскалаторе был только i -ый человек, то его время прибытия было равно $t_i \cdot w_i$. Но у него есть помеха – тот, кто идёт перед ним. Тогда итоговое время будет $T_i = \max\{T_{i-1}, t_i \cdot w_i\}$. И рассматривать людей следует снизу вверх.

Осталось дело за малым – вести полученные результаты в нужном порядке. В решении на C++ снова применяется сортировка, а на Паскале (так как менялись номера людей, а не сами данные) всё выводится по порядку.

Решение на языке C++

```
1  #include <iostream>
2  #include <cstdio>
3
4  #include <vector>
5  #include <algorithm>
6
7  #define MAX(a,b) ((a)>(b)?(a):(b))
8
9  using namespace std;
10
11 typedef long long LL;
12
13 bool cmp(pair<pair<int,int>,pair<int,LL> > a,
14         pair<pair<int,int>,pair<int,LL> > b){
15
16     return a.first.second < b.first.second;
17 }
18 bool num(pair<pair<int,int>,pair<int,LL> > a,
19         pair<pair<int,int>,pair<int,LL> > b){
20
21     return a.second.first < b.second.first;
22 }
23
24 int main(){
25     freopen("input.txt", "r", stdin);
26     freopen("output.txt", "w", stdout);
27
28     int n;
29     vector<pair< pair<int,int>, pair<int,LL> > > v;
30     cin>>n;
31     v.reserve(n);
32     for(int i=0;i<n;++i){
33         int a, b;
34         cin>>a>>b;
35         v.push_back(make_pair(make_pair(a,b), make_pair(i, a*1LL*b)));
36     }
37     sort(v.begin(), v.end(), cmp);
38
```

```
39     for(int i=1;i<n;++i){
40         v[i].second.second = MAX(v[i].second.second, v[i-1].second.second);
41     }
42
43     sort(v.begin(), v.end(), num);
44
45     for(int i=0;i<n;++i) cout<<v[i].second.second<<endl;
46
47     return 0;
48 }
```

Решение на языке Паскаль

```
1  uses math;
2  var
3      n, i : Longint;
4      mm : array [0..100500,0..1] of Longint;
5      tmp : Int64;
6      res : array [0..100500] of Int64;
7      m : array [0..100500] of Longint;
8
9  procedure qs(l, r : Longint);
10 var
11     i, j, x, t : Longint;
12 begin
13     i := l;
14     j := r;
15     x := mm[m[l + random(r-l)], 1];
16     repeat
17         while mm[m[i], 1] < x do inc(i);
18         while mm[m[j], 1] > x do dec(j);
19         if i <= j then begin
20             t := m[i];
21             m[i] := m[j];
22             m[j] := t;
23             inc(i);
24             dec(j);
25         end;
26     until i > j;
27     if l < j then qs(l, j);
28     if r > i then qs(i, r);
29 end;
30
31 begin
32     randomize;
33     assign(input, 'input.txt'); reset(input);
34     assign(output, 'output.txt'); rewrite(output);
35
36     Read(n);
37     tmp := 1;
38     for i:=0 to n-1 do begin
39         Read(mm[i, 0], mm[i, 1]);
40         m[i] := i;
41         res[i] := tmp * mm[i, 0] * mm[i, 1];
42     end;
43     qs(0, n-1);
44     for i:=1 to n-1 do
45         res[m[i]] := max(res[m[i]], res[m[i-1]]);
46
47     for i:=0 to n-1 do WriteLn(res[i]);
48 end.
```