AtCoder Beginner Contest 113 解説

writer: namonakiacc, square1001, E869120

平成 30 年 11 月 4 日

A: Discount Fare

鉄道の運賃が X 円、バスの運賃が Y 円のとき、A 駅から C 駅まで移動するのに $X + Y \div 2$ 円かかります。

よって、$X + Y \div 2$ を出力します。

```
#include <cstdio>
int X,Y;
int main()
{
	scanf("%d%d",&X,&Y);
	printf("%d\n",X+Y/2);
}
```

B: Palace

地点 i の平均気温は $T - H_i \times 0.006$ 度なので、各地点の平均気温を計算し、それらの中から平均気温が A 度に最も近い点を求めるとよいです。

気温は整数ではありませんが、気温を 1000 倍して計算すると浮動小数点型を用いなくても答えを求めることができます。

```
#include <cstdio>
int N,T,A,X;
int main()
{
	scanf("%d%d%d",&N,&T,&A);
	int res=0;
	int cc=1<<30;
	for(int i=1;i<=N;i++)
	{
		scanf("%d",&X);
		int d=(T*1000-X*6)-A*1000;
		if(d<0)d=-d;
		if(cc>d)cc=d,res=i;
	}
	printf("%d\n",res);
}
```

C: ID

県ごとに、その県に属する市を誕生した順にソートして、認識番号を割り振るとよいです。

```
#include <cstdio>
#include <vector>
#include <algorithm>
using namespace std;
typedef long long ll;
#define rep(i,n) for(int i=0;i<n;i++)

int N,M;
int P[100000],Y[100000];
vector<int> yd[100001];

int main()
{
	scanf("%d%d",&N,&M);
	rep(i,M)scanf("%d%d",&P[i],&Y[i]),yd[P[i]].push_back(Y[i]);
	rep(i,N)sort(yd[i+1].begin(),yd[i+1].end());
	rep(i,M)printf("%012lld\n",ll(P[i])*1000000+int(lower_bound(yd[P[i]].
	begin(),yd[P[i]].end(),Y[i])-yd[P[i]].begin())+1);
}
```

D:　Number of Amidakuji

$H=100, W=8$ のケースで、あみだくじの個数は $34^{100} \fallingdotseq 1.4*10^{153}$ 個程度あり、到底全探索できそうにありません。そこで、「動的計画法」という方法を使ってみることにしましょう。

$A(h,x)$ を、縦から h 列目を通過した (横棒を通った場合も含む) 直後に左から x 本目の縦棒にいる場合の数とします。$h=0$ のとき、1 列目より前を表すことにます。

$h=0,1,2,3,\cdots,H$ の順に上から下に降りていく感じで $A(h,x)$ を更新していくことにします。左から 1 本目の縦棒からスタートするので、$A(0,1)=1$、$A(0,2)=A(0,3)=\cdots=A(0,W)=0$ になります。

$A(h,x)$ が指しているあみだくじの経路からは、$A(h+1,x-1), A(h+1,x), A(h+1,x+1)$ のいずれかの状態に行くことになります。$A(h,x)$ の状態から何通りが $x-1$ 本目の縦棒に行って、何通りが x 本目の縦棒にとどまって、何通りが $x+1$ 本目の縦棒に行くかは、$h+1$ 本目の横棒の配置を全探索することで計算することができます。先ほどの $x-1, x, x+1$ 本目に行く場合の数をそれぞれ X, Y, Z とおくと、$A(h+1,x-1), A(h+1,x), A(h+1,x+1)$ の結果はそれぞれ $X*A(h,x), Y*A(h,x), Z*A(h,x)$ 足されることになります。

したがって、計算量 $O(HW2^W)$ で解くことができます。$H=100, W=8$ のケースでも、十分に間に合わせることができます。

おまけ: 実は先ほどの X,Y,Z の値はフィボナッチ数列 ($F_1=F_2=1, F_N=F_{N-1}+F_{N-2} (N \geq 3)$ を満たす数列) を使って求めることができます。ある 1 列の横棒の配置の場合の数は、フィボナッチ数 F_{W+1} です。これを使えばこの問題を計算量 $O(HW)$ で解くことができます。ぜひ考えてみましょう！

サンプルコード (C++)：
`https://beta.atcoder.jp/contests/abc113/submissions/3540482`