AtCoder Beginner Contest 034
解説

AtCoder株式会社 代表取締役
高橋 直大

競技プログラミングを始める前に

AtCoder

- 競技プログラミングをやったことがない人へ
  - まずはこっちのスライドを見よう！
  - http://www.slideshare.net/chokudai/abc004

A問題　テスト

A問題　問題概要

- 整数X, Yが与えられる。
- X<YならBetter、そうでないならWorseを出力しなさい

- 制約
  - $0 \leqq X, Y \leqq 100$
  - $X \neq Y$

- 基本的なプログラムの流れ
  - 標準入力から、必要な入力を受け取る
    - 今回の場合は、X, Y という2つの整数
  - 問題で与えられた処理を行う
    - 今回は、X<Yの大小関係の判定
  - 標準出力へ、答えを出力する

- 入力
  - 2つの整数を、標準入力から受け取る
    - Cであれば、scanf(“%d %d”, &X, &Y);　など
    - C++であれば、cin >> X >> Y;
    - 入力の受け取り方は、下記の練習問題に記載があります。
      - http://practice.contest.atcoder.jp/tasks/practice_1

- 今回の問題は、X, Yの大小関係を出力する
- X, Yの大小関係ってどう求めればいいの？
  - 不等号で判定して、If ～ else文で分岐しちゃいましょう。
  - 例えばこんな感じ

```
string ans;
if(X < Y) ans = “Better”;
else ans = “Worse”;
```

A問題　アルゴリズム

AtCoder

- 出力
  - 求めた答えを、標準出力より出力する。
  - 言語によって違います。
    - printf(“%s¥n”, ans); (C)
    - cout << ans << endl; (C++)
    - System.out.println(ans); (Java)
    - 各言語の標準出力は、下記の練習問題に記載があります。
      - http://practice.contest.atcoder.jp/tasks/practice_1

B問題　ペア

B問題　問題概要

- 10^9 人の人がいます。人には 1 から 10^9 までの番号がついています。
- 1 番と 2 番の人、3 番と 4 番の人、5 番と 6 番の人、… がペアになりました。
- n 番の人とペアになった人の番号を求めてください。

- 制約
    - $1 \leqq n \leqq 10^9$

- ペアを順番に作る・・・・？
  - N=9が与えられた！
    - (1,2)がペア
    - (3,4)がペア
    - (5,6)がペア
    - (7,8)がペア
    - (9,10)がペア
      - 9番とペアな人は10番だ！
  - 一見これでも解けそう

- ペアを順番に作る‥‥？
  - N=999999999が与えられた！
    - (1,2)がペア
    - (3,4)がペア
    - (5,6)がペア
    - (7,8)がペア
    - (9,10)がペア
    - ……
    - ……
    - 時間切れ
  - 工夫しないと解けない＞＜

- ペアの法則を見つけよう！
  - N=1 の時、答えは2
  - N=2 の時、答えは1
  - N=3 の時、答えは4
  - N=4 の時、答えは3
  - N=5 の時、答えは6
  - N=6 の時、答えは5
- 法則は簡単！
  - Nが偶数の時答えがN-1
  - Nが奇数の時答えがN+1

B問題　アルゴリズム

- どうやってコードにしよう？
  - Nが奇数であるかどうかの判定
    - A % B で、「AをBで割った余り」を求められる。
      - 言語によっては%の代わりにmodだったりするから調べよう！
    - if(N%2==1) で奇数判定出来る！
      - C言語などはif(N%2)だけでも良い。
  - これを使えば、答えの求め方は簡単！

```
int N ← input
int ans;
if(N%2 == 1) ans = N + 1;
else ans = N – 1;
output → ans
```

- bit演算あれこれ。
  - If((N&1)==1)で奇数判定が出来る
    - Nを二進数にした時に、1bit目が1であれば1、そうでなければ0
  - (((N-1)^1)+1)を出力してもいい
    - (1,2)(3,4)を(0,1)(2,3)に置き換えるためにまず-1
    - (0,1)(2,3)などのペアは、1bit目を反転させたものなので^1
    - (1,2)(3,4)の関係性に戻すために+1
  - (N-1)+(N&1)*2とかでも良い。
    - 奇数の時はN&1が1になるので、偶数のパターンと比べて答えが2増えるのを利用する。

  - 特にこういう回答を書く必要はない。

C問題　経路

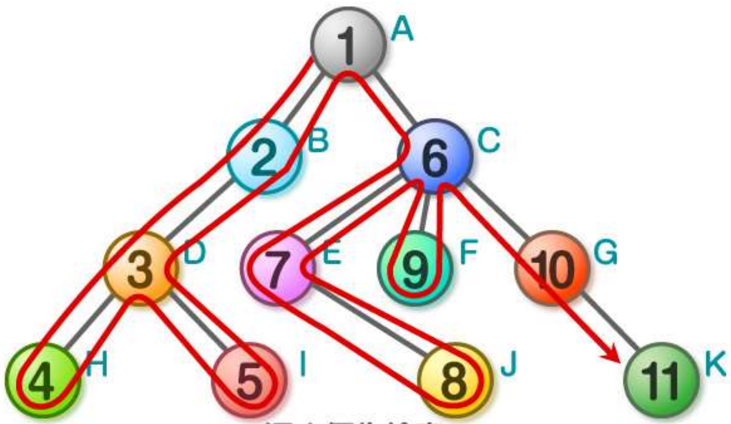
C問題　問題概要

- 高橋君は、マス目 (i,j) から $(i+1,j)$ または $(i,j+1)$ に進むことができます。
- 高橋君が (1,1) から (W,H) まで行く経路の個数を 1,000,000,007 で割ったあまりを求めてください。

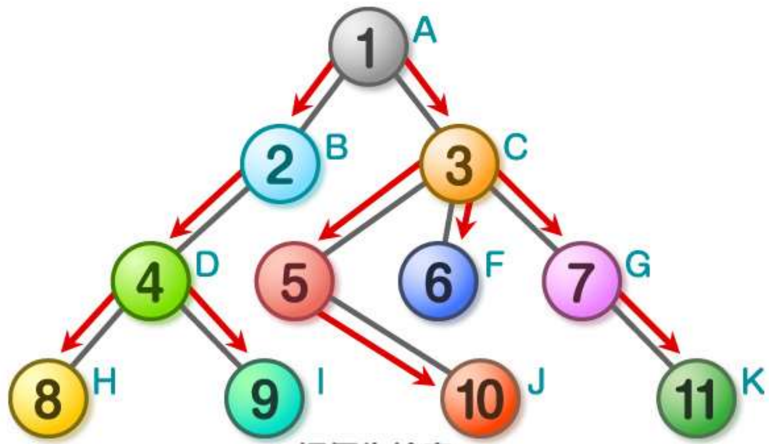
- 制約
  - 1 ≦ W,H ≦ 100,000

C問題　アルゴリズム

- 経路の数え方
  - 探索を使う？
    - 深さ優先探索
    - 幅優先探索
  - 確かに経路個数は数えられる
    - でも時間が足りない！
    - 50点の部分点は得られる

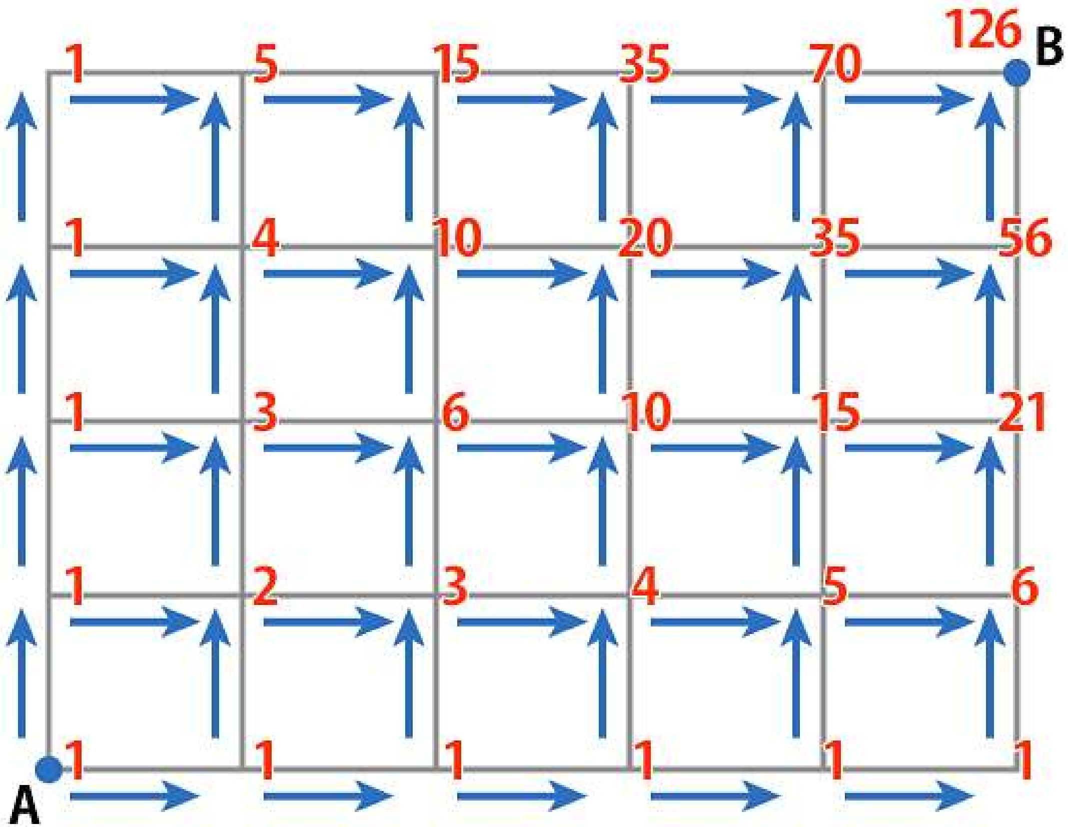


深さ優先検索



幅優先検索

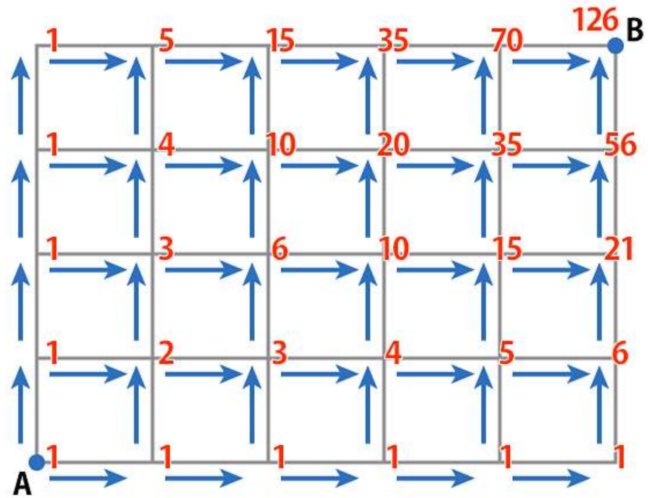
- 経路の数え方
  - 動的計画法？



- 経路の数え方
  - 各地点に辿り着く経路数をメモする
  - 周りのマスから、次のマスへの経路数をメモする
  - こうすると、計算するべき地点はW×H個しかない！

  - これで100点
    - 101点は取れない
  - 実装は2次元配列などで！

```
DP[a,b] = DP[a-1,b]+DP[a,b-1];
DP[a,b] %= mod;
```



- 数学的に計算しよう！
  - 問題を読み替えると、「上にH-1回、右にW-1回移動する組み合わせの個数」を求めれば良い。
  - これは、(H+W-2)回移動する中から、H-1回移動する方法を求めれば良い！
    - これは、${}_{H+W-2}C_{H-1}$で求まる！
    - つまり、(H+W-2)! / (H-1)!(W-1)!で求まる。

- (H+W-2)! / (H-1)!(W-1)!の求め方
  - 普通にforループを求めれば出来そう？
  - 問題は、「 mod = 1,000,000,007 で割った余りを求めなさい」
    - (H+W-2)!や(H-1)!(W-1)!は大きすぎて計算しきれない
    - そこで、(H+W-2)!をmodで割ってあまりを計算してしまうと、普通に割り算は出来ない！
  - どうすればいいか？

- 逆元を求めよう！
  - M = 1,000,000,007は素数なので、mod Mでの割り算は、掛け算に変換できる！
  - フェルマーの小定理により、
    - pが素数の時、$a^{p-1} \equiv 1 \ (mod \ p)$
    - つまり、$a^{p-2} \equiv a^{-1} (mod \ p)$
    - これを利用して、Aで割りたい時は、AのM-2乗で掛ければ良い！
      - 累乗a^b (mod p)の計算は、実はO(logb)で出来る！
        - » 例えば、32乗を計算したい時は、16乗を計算して、その2つをかける。
        - » 33乗を計算したい時は、32乗から1乗をかける。

      - ぶっちゃけ初めての人は知らなくて当たり前なので、これを機会に覚えよう！

C問題　アルゴリズム

- 累乗a^b (mod p)の計算の仕方！
  - » 解りやすい計算方法を紹介するので、早いのは自分で調べよう！
- aのb乗をpで割った余りを計算する関数
  - » オーバーフローは無視して描いてます。
  - – calc(int a, int b, int p)について考える
    - b==0の時、
      - – return 1;すれば良い。
    - bが偶数の時、
      - – Int d = calc(a, b/2, p);
      - – return (d * d) % p; とすると、bが半分に出来る。
    - bが奇数の時、
      - – return (a * calc(a, b-1, p)) % p;　とすれば良い。
  - – 奇数の後は偶数が必ず呼ばれるので、2回に1回はbが半分になる。よって計算量はO(logb)

D問題 食塩水

D問題　問題概要

- 食塩水が入った容器が N 個あります。容器には 1 から N までの番号がついています。
- i 番の容器には濃度 p_i % の食塩水が w_i グラム入っています。
- 高橋君は、K 個の容器を選び、選んだ容器の中に入っている食塩水をすべて混ぜ合わせることにしました。高橋君の混ぜた食塩水の濃度として考えられる最大値を求めてください。

- 制約
  - 1≦K≦N≦1,000
  - 1≦w_i ≦10^9
  - 0≦p_i ≦100

D問題　アルゴリズム

- 直接答えを求める事が出来るか？
  - 例えば、N=4、K=2で、30%, 25%, 20%, 10%の食塩水があったとする。
  - どれを使うといいかが解るか？
    - 30%は使いそう
    - 25%も使いそうだけど、状況によっては使わない
      - 30%が100リットル、25%が100リットル、20%が1リットルだったら、20%を使った方がいい。
    - 20%もたまに使いそう
    - 10%でも使う時は使う
  - 良くわからない！！！
    - このままだと、どれを使っていいのかが解らない

AtCoder

- 「どれを使うべき」という優先順位を付けたい
  - どうすればいいのか？
  - 目標とするパーセンテージを決めてしまえば良い！
    - こうすることにより、「どの容器がどれだけのメリットがあるか、というのが解る」
  - 例えば入力例1で22%を目指す場合、
    - 入力：　A: (100,15), B: (300,20), C: (200,30)
    - Aを入れる
      - 22%を目指したいのに15%しかない。7%足りない。
      - 7%足りない液体を100g入れる　→ 7g分食塩が足りない
    - Bを入れる: 300 * (0.20 – 0.22) = 6　→　6g分食塩が足りない
    - Cを入れる: 200 * (0.30 – 0.22) = 16　→　16g分食塩が余る
    - つまり、C,B,Aの優先度で混ぜていけばいい！

- 目標のパーセンテージをどう決めるか？
  - A%以上の食塩水を作れるなら、A-1%以上の食塩水も絶対作れる！
    - 同じものを作ればいいので当たり前。
    - これは、さっきの優先度の決め方でも当然作れる。
      - そのパーセンテージを達成するために最善の方法を選ぶため。
  - だったら、「作れるライン」「作れないライン」を見極めれば良い！
  - こういう時は、二分法を使おう！

D問題　アルゴリズム

AtCoder

- 二分法とは？
  - 「絶対に作れるライン」と「絶対に作れないライン」の2つを設定します！
    - Double OK = 0
    - Double NG = 100
  - 間を決めます。
    - Double Mid = (OK + NG) / 2;
  - これについて判定し、OKとNGを書き換えます。
    - If(check(mid)) OK = Mid;
    - Else NG = Mid;
  - これを繰り返すことにより、OKとNGの差を半分ずつにすることが出来る！
    - 100回もやれば誤差が殆どなくなります！

D問題　アルゴリズム

- こうした、「平均の最大化問題」を解くには、「答えを先に決めてしまう二分法」が非常に有効になります！
  - ICPCなどでもよく出題されている、競技プログラミングの鉄板アルゴリズムの一つです。

D問題　アルゴリズム

AtCoder

- 貪欲法で通した方へ
  - これで落ちるらしいです。

    4 3

    16 51

    51 64

    61 57

    4 26

    答えは59.039062499999993くらい。