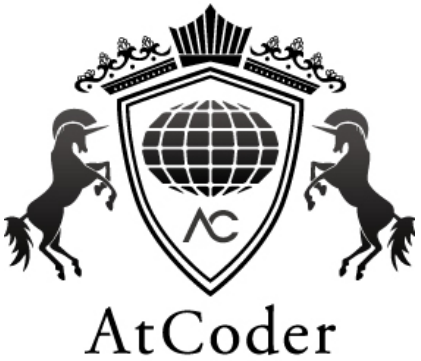
実践・最強最速のアルゴリズム勉強会
第一回　講義資料



AtCoder株式会社 代表取締役
高橋 直大

はじめに！

AtCoder

- 本講義では、ソースコードを扱います。
- 前面の資料だけでは見えづらいかもしれないので、手元にファイルを置いておきましょう。
- URLはこちらから
  - http://www.slideshare.net/chokudai/WAP_AtCoder1
  - URLが打ちづらい場合は、Twitter: @chokudaiの最新発言から飛べるようにしておきます。

目次

1. 勉強会の流れ
2. 競技プログラミングって？
3. シミュレーション問題
4. 全探索問題
5. 本日のまとめ

勉強会の流れ

1. 勉強会の日程
2. 1日の流れ

勉強会の日程

AtCoder

1日目　シミュレーションと全探索

2日目　色々な全探索

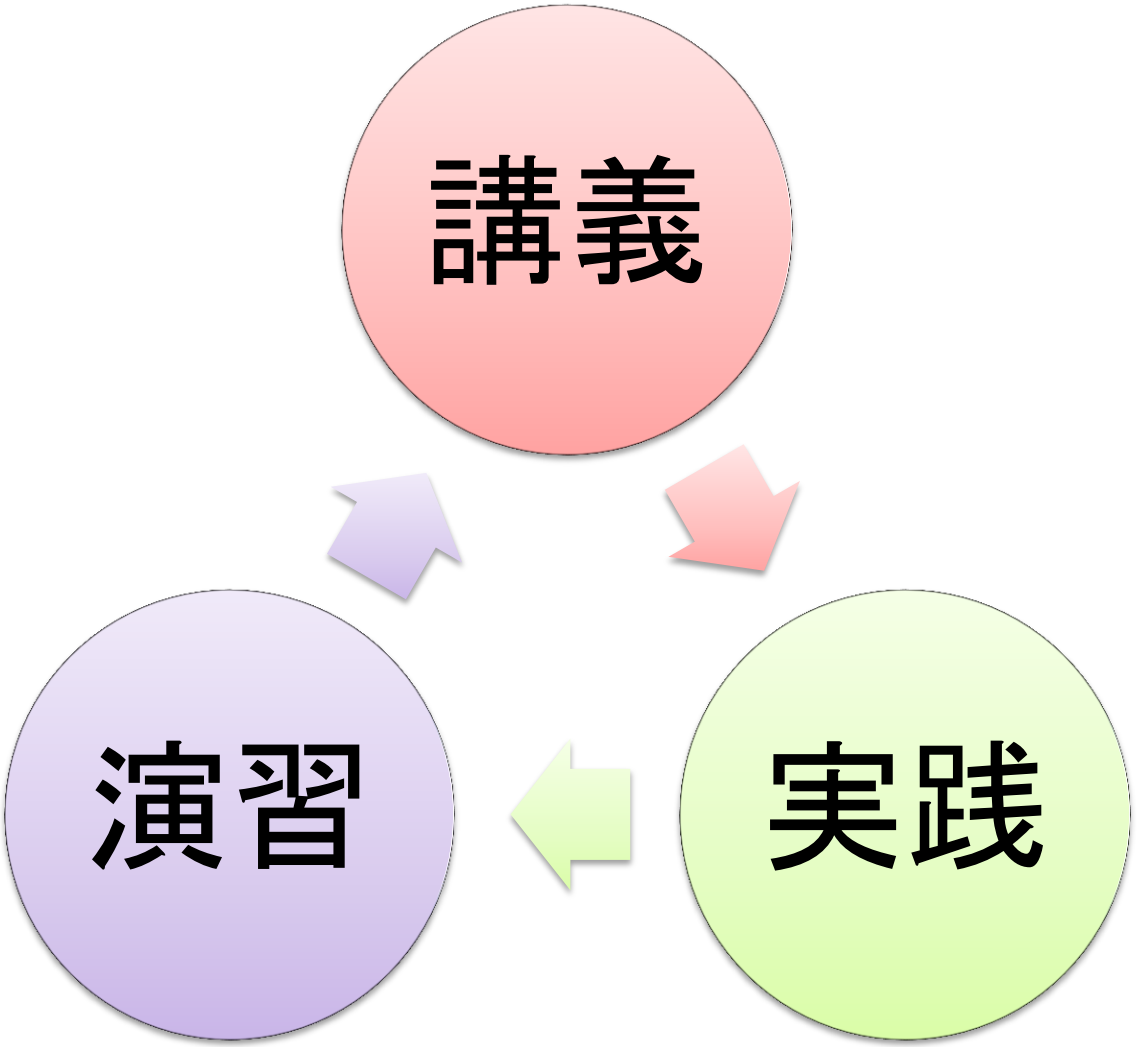
3日目　動的計画法とメモ化再帰

4日目　動的計画法と計算量を減らす工夫

5日目　難問に挑戦！

１日の流れ

AtCoder



１日の流れ

AtCoder

講義

- 基礎的なアルゴリズムを学ぶ
- 必要な知識を補う

実践

- 実際に問題例を見る
- コードでの表現方法を覚える

演習

- 自分で問題を解いてみる！
- コードを書いて理解を深める

１日の流れ

AtCoder

- 演習について
  - 実力差があると思うので、暇な時間が出来る人、ついていけない人、出ると思います。
    - どうしようもないので、早い人は支援に回ってもらえると嬉しいです！
  - 解らないことがあったら、#WAP_AtCoderでTwitterに投稿！
    - 多分早く終わった人が質問回答してくれます。
    - 具体例はこんな感じ
      - 「今やってる問題どれですか！」
      - 「この解答のWAが取れません！　http:// ~~」
      - 「コンパイルエラー出るよーなんでー＞＜　http:// ~~」
      - とりあえずコードが書けてたら、間違ってても提出してURLを貼りつけよう
  - もちろん、手を上げて質問してくれてもOK!
    - 回りきれる範囲では聞きに行きます。

競技プログラミングって？

1. 競技プログラミングとは何か？
2. 競技プログラミングで身に付くアルゴリズム
3. 今日学ぶこと

競技プログラミングとは何か？

AtCoder

- アルゴリズムを考えるプログラミングコンテストのためのプログラミング

- じゃあ、プログラミングコンテストって？

競技プログラミングとは何か？

AtCoder

- プログラミングコンテストの種類

テーマが決められていて、自由にプログラムを作る競技	• これが主流ですが、今回これは取り扱いません。
作るべきプログラムが決められていて、早く正確に組む競技	• 75分～5時間程度の短めの期間 • 与えられる問題は3～10問程度
作るべきプログラムが決められていて、完成度を競う競技	• 1週間～数か月と長めの期間 • 与えられる問題は1問

競技プログラミングとは何か？

AtCoder

- プログラミングコンテストの種類

テーマが決められていて、自由にプログラムを作る競技	• これが主流ですが、今回これは取り扱いません。
作るべきプログラムが決められていて、早く正確に組む競技	• 75分～5時間程度の短めの期間 • 与えられる問題は3～10問程度
作るべきプログラムが決められていて、完成度を競う競技	• 1週間～数か月と長 • 与えられる問題は1問

今回はこれ！

競技プログラミングって？

AtCoder

- 全体の流れ
  1. 問題文を読む
  2. 問題で要求されたプログラムを書く
  3. ジャッジサーバーにプログラムを提出する
  4. 提出結果がAcceptになったら正解！
  5. 制限時間内にたくさんの問題に正解した人の勝ち！

競技プログラミングって？

AtCoder

- まずは問題一覧をチェック！
  - 上の問題のほうが簡単であることが多いよ！

AtCoder Regular Contest #002 2012/05/02 21:00:00 ~ 2012/05/02 22:30:00

トップページ | 問題 | 提出 | 質問 | 結果 | 順位表

Tasks

#	問題名	時間制限	スタック制限	メモリ制限	
A	うるう年	2 sec	10 MB	64 MB	提出する / 問題を編集する
B	割り切れる日付	2 sec	10 MB	64 MB	提出する / 問題を編集する
C	コマンド入力	2 sec	10 MB	64 MB	提出する / 問題を編集する
D	ボードゲーム	2 sec	10 MB	256 MB	提出する / 問題を編集する

競技プログラミングって？

AtCoder

- 問題文を見よう！

AtCoder Regular Contest #002 2012/05/02 21:00:00 ~ 2012/05/02 22:30:00

トップページ | 問題 | 提出 | 質問 | 結果 | 順位表

A - うるう年

時間制限：2sec / スタック制限：10MB / メモリ制限：64MB

問題文

高橋君は忘れっぽい性格なので、うるう年は 2 月 29 日の存在を毎回忘れてしまいます。
そこで、自動でうるう年かどうかをコンピュータに教えてもらえるようにしたいと思います。
入力として与えられた年がうるう年かそうでないかを判断しなさい。

ただし、うるう年は以下の規則で決定します。

- 規則 1:4 で割り切れる年はうるう年である。
- 規則 2:100 で割り切れる年をうるう年ではない。
- 規則 3:400 で割り切れる年はうるう年である。
- 規則 4:規則 1～3 のいずれも満たさない場合は，うるう年ではありません。

ただし、規則 1 ～ 3 の内に複数満たすものがあれば後の規則(数字の大きな規則)が優先されます。例えば、2000 年は規則 3 を満たすのでうるう年です。

競技プログラミングって？

AtCoder

- 問題文を見よう！
- 問題文の要素は以下の４つ
  - 問題文
    - どういう問題を解くのかの説明
  - 入力
    - どういった入力が与えられるのかの説明
    - 入力のフォーマットや、値の上限・下限など
  - 出力
    - 何をどういったフォーマットで出力するべきか
  - 入出力例
    - 実際に与えられる入力・出力の例

競技プログラミングって？

AtCoder

- プログラムを書こう！
  - ここは、問題によって書くものが全然違います。

```java
import java.util.*;

public class Main{
    public static void main(String[] args){
        new Main().run();
    }

    void run()
    {
        //標準入力にはScannerを使います
        Scanner cin = new Scanner(System.in);

        //標準入力から、整数nを受け取ります。
        int n = cin.nextInt();

        //答えを入れておく変数を作ります。
        String ret = "";

        //偶奇の判定を行います。
        if(n % 2 == 0) ret = "Blue";
        else ret = "Red";

        //答えを出力します。
        System.out.println(ret);
    }
}
```

競技プログラミングって？

AtCoder

- プログラムを提出しよう！

競技プログラミングって？

AtCoder

- 結果を確認しよう！

ここが緑になればOK！

競技プログラミングって？

AtCoder

- 結果を確認しよう！
  - 緑のACが出たら正解！
  - オレンジが出ると不正解。
    - CE コンパイルエラー(Compile Error)
      - 詳細を開いて、コンパイルエラーがどこで発生したか確認しよう！
    - RE ランタイムエラー(Runtime Error)
      - 実行時エラーが起きてしまった場合。バグがあるので直そう！
    - WA 誤解答(Wrong Answer)
      - 答えが間違っていた場合。バグがあるか、アルゴリズムがおかしい
    - TLE 制限時間オーバー(Time Limit Exceeded)
      - 制限時間がオーバーしていた場合。実行時間が遅いプログラムを書いてしまっているか、バグで必要以上にループしてしまっている？
    - 他にもあるけど、気を付けるべきは殆どこれだけ！

競技プログラミングって？

- ランキングを確認しよう！
  - コンテストの時限定。今回の勉強会では関係ないです。

AtCoder Regular Contest #014 2013/06/16 20:00:00 ~ 2013/06/16 21:30:00 AtCoderへ戻る 日本語 chokudai@AtCoder社長(guest)

トップページ 問題 提出 質問 結果 順位表

順位表

順位	ユーザ名	君が望むなら世界中全てのたこ焼きを赤と青に染め上げよう	あの日したしりとりの結果を僕達はまだ知らない。	魂の還る場所	grepマスター	得点 / Total
1	FtnCky	100 00:59	100 04:45	100 13:53	100 24:23	400 24:23
2	アルミ缶の上にある民家	100 04:01	100 03:09	100 09:28	100 29:38	400 29:38
3	Kirino :P	100 03:14	100 08:51	100 14:45	100 31:32	400 31:32
4	kawatea	100 00:52	100 05:05	100 17:36	100 33:39	400 33:39
5	tmt514	100 01:32	100 (1) 12:14	100 18:11	100 28:58	400 (1) 33:58
6	natsugiri	100 01:12	100 (1) 07:14	100 14:48	100 31:06	400 (1) 36:06
7	ടന‿ke	100 01:50	100 06:43	100 11:49	100 37:59	400 37:59
8	sune2	100 01:05	100 05:06	100 13:11	100 39:37	400 39:37
9	レインボーダッシュ(会場)	100 01:18	100 07:47	100 41:04	100 30:42	400 41:04
10	まーす	100	100 (1)	100	100	400 (1)

競技プログラミングって？

AtCoder

- ランキングを確認しよう！
  - 正解数の多い人が上位
  - 正解数が同じ場合は、提出の早い人が上位

競技プログラミングで身に付くアルゴリズム

AtCoder

- 通常のアルゴリズム教育
  - アルゴリズムの知識を網羅的に学ぶ
  - 知識としては知っていても、実際に使用し辛い
- 競技プログラミングで身に付くアルゴリズム
  - 基礎的なアルゴリズムが殆ど
    - しかし、簡単なわけではない
  - 基礎的なアルゴリズムを、以下に実践的な場面で上手に使うかが身に付く！
  - そんなに多くのアルゴリズムを覚えるわけではない。

今日学ぶこと

AtCoder

- 今日学んで帰ることは、以下の2つ！
  - シミュレーション
    - ○○をしなさい、という形式の課題
    - ○○をする操作を愚直に実装する
  - 全探索
    - ○○のうち、最も良いものを出力しなさい、という課題
    - 与えられた選択肢を愚直に全て試す
- 最も基礎的だけれども、最も重要なアルゴリズム！

シミュレーション問題

1. シミュレーション問題って？
2. 最も単純なシミュレーション問題
3. 繰り返しの発生するシミュレーション問題
4. 難しいシミュレーション問題（おまけ）

シミュレーション問題って？

AtCoder

- シミュレーション問題とは？
  - 言われたことを素直にシミュレーションすれば、答えが求まる問題！
  - 問題文に書かれていることを、そのまま実装してあげれば良い
    - 難易度の高い問題だと、そう簡単に行かないこともある。

シミュレーション問題って？

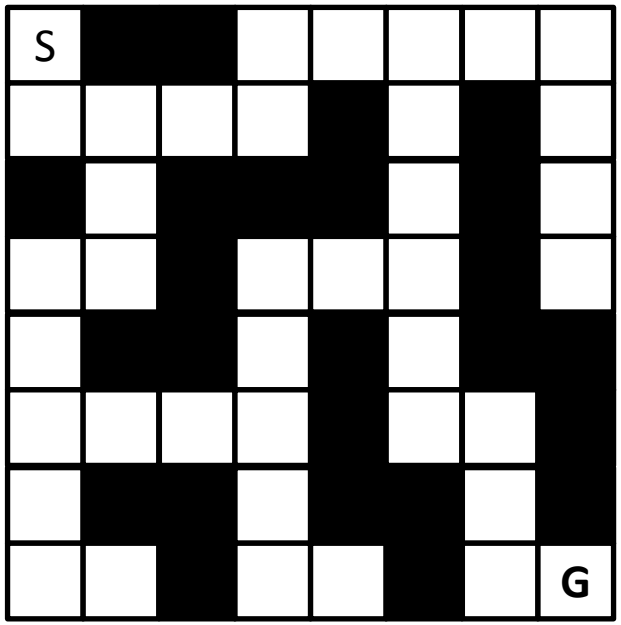
AtCoder

- シミュレーション問題とその他の問題の違い
  - シミュレーション問題
    - 何をすれば良いかが明確に書かれている。
    - 書かれていることをそのまま実装すれば良い。
  - その他の問題
    - したいことは書かれているが、それを達成するために、必要なことが全て書かれているわけではない。
    - どのように実装するかは、自分で考えなくてはいけない。

シミュレーション問題って？

AtCoder

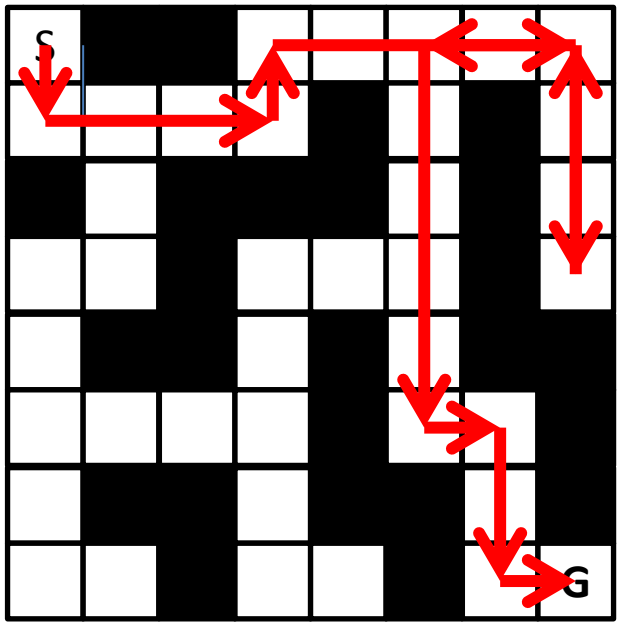
- 迷路を使った具体例
  - シミュレーション問題の場合
  - Sから左手をついて移動していった時に、ゴールまでたどり着くのに何マス移動が必要か答えなさい。



シミュレーション問題って？

AtCoder

- 迷路を使った具体例
  - シミュレーション問題の場合
  - Sから左手をついて移動していった時に、ゴールまでたどり着くのに何マス移動が必要か答えなさい。

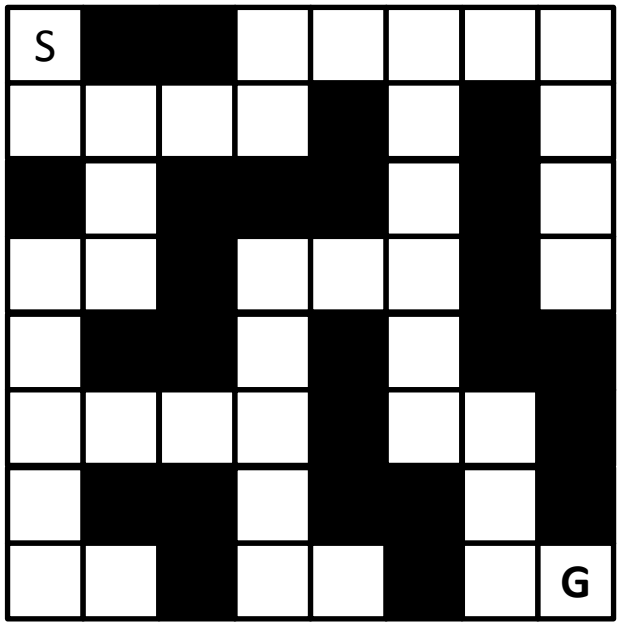


やるべきことは
一直線！

シミュレーション問題って？

AtCoder

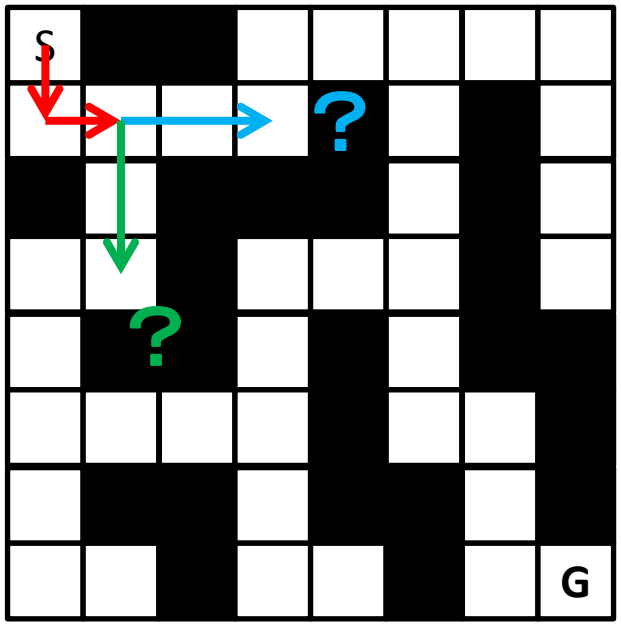
- 迷路を使った具体例
  - その他の問題の場合
  - SからGまでの最短経路を求めなさい。



シミュレーション問題って？

AtCoder

- 迷路を使った具体例
  - その他の問題の場合
  - SからGまでの最短経路を求めなさい。



選択肢が複数ある

何をすれば良いかを、
自分で考えないとダメ

シミュレーション問題って？

AtCoder

- シミュレーション問題
- 最も単純なシミュレーション問題
  - ○○をしなさい。系問題
  - ただの足し算・掛け算・ifによる分岐で出来る
- 繰り返しのあるシミュレーション問題
  - ○○を何回しなさい。系問題
  - Forループや配列などが必要になることも？

- どちらにしても、言われた操作をするだけ！
  - これが出来るのが、競技プログラミングをする上で大前提となる。

最も単純なシミュレーション問題　実践

AtCoder

- ARC014 A問題　君が望むなら(略
- http://arc014.contest.atcoder.jp/tasks/arc014_1

- 問題概要
  - 整数Nが与えられる。
  - Nが奇数なら”Red”、偶数なら”Blue”と出力しなさい。

最も単純なシミュレーション問題　実践

AtCoder

- やるべきこと
  - 入力の受け取り
    - 標準入力から、整数Nを読み込む
  - 問題で指示された処理
    - 今回の場合は、偶奇の判定
  - 回答の出力
    - 標準出力に、回答を出力する

最も単純なシミュレーション問題　実践

AtCoder

- 標準入力のやり方
  - Javaの場合は、Scannerを使おう！
    - それ以外の人は各自ググってください！
- 偶奇の判定
  - %を使うと余剰の判定が出来ます。
  - If (n % 2 == 0) みたいな感じで使います。
    - 一部言語では違うと思うので、その場合は調べてね！
- 標準出力の仕方
  - Javaであれば、System.out.printlnとかで良いです。
  - 他もググればすぐに出てくると思います。
- 標準入力、および標準出力が解らない場合は、以下のURLをチェック
  - http://practice.contest.atcoder.jp/tasks/practice_1

最も単純なシミュレーション問題　実践

AtCoder

- http://arc014.contest.atcoder.jp/submissions/140808

```java
import java.util.*;

public class Main{
    public static void main(String[] args){
        new Main().run();
    }

    void run()
    {
        //標準入力にはScannerを使います
        Scanner cin = new Scanner(System.in);

        //標準入力から、整数nを受け取ります。
        int n = cin.nextInt();

        //答えを入れておく変数を作ります。
        String ret = "";

        //偶奇の判定を行います。
        if(n % 2 == 0) ret = "Blue";
        else ret = "Red";

        //答えを出力します。
        System.out.println(ret);
    }
}
```

最も単純なシミュレーション問題　演習

AtCoder

- ABC004 A問題
- http://abc004.contest.atcoder.jp/tasks/abc004_1

- 問題概要
  - 整数Nが与えられる。
  - 整数Nを2倍した数字を出力しなさい。

最も単純なシミュレーション問題　演習

AtCoder

- 早くできた人は、この問題も解いてみよう！
  - 軽く触れますが、解説はしません。
- ABC001 B問題　視程の通報
  - http://abc001.contest.atcoder.jp/tasks/abc001_2
- ABC001 C問題　風力観測
  - http://abc001.contest.atcoder.jp/tasks/abc001_3

最も単純なシミュレーション問題　演習

AtCoder

- 標準入力から数字を読み取る！
  - さっきと同じ
- 貰った入力を2倍にする！
  - N = 2 * N;　とかでOK
  - 個人的には、「入力された数字は弄らない」とした方が、規模が大きくなってきた時にバグが出にくいと思ってます。
    - なので、int ret = 2 * N;とかを推奨します。
- 標準出力から答えを出力する！
  - さっきと同じ

最も単純なシミュレーション問題　演習　解説

AtCoder

- ソースコード
  - http://abc004.contest.atcoder.jp/submissions/140820

```java
import java.util.Scanner;

public class Main{
    public static void main(String[] args){
        new Main().run();
    }

    void run()
    {
        //標準入力にはScannerを使います
        Scanner cin = new Scanner(System.in);

        //標準入力から、整数nを受け取ります。
        int n = cin.nextInt();

        //答えを求めます。
        int ret  = n * 2;

        //答えを出力します。
        System.out.println(ret);
    }
}
```

最も単純なシミュレーション問題　演習

AtCoder

- 難しい問題になると・・・？
  - やることがものすごく増える！
  - ABC001 B,Cの問題を見れば、どれだけ面倒になるかがわかる。
    - ABC001 B問題　視程の通報
      - http://abc001.contest.atcoder.jp/tasks/abc001_2
    - ABC001 C問題　風力観測
      - http://abc001.contest.atcoder.jp/tasks/abc001_3
  - 正直な話、面倒な処理が増えるだけで、あまり面白味のある問題ではない。
    - 学習目的で得られるものもあまり多くない。
    - よって、コンテストでもあまり出題されない。

最も単純なシミュレーション問題　攻略法！

AtCoder

- If文を高速に書けるようになろう！
  - 基本的には、if文での分岐が書けるかが大切

- 「書かれていること」を実装できるようにしよう！
  - 「するべきこと」は解るけれども、「どうするべきか」が解らない！となってしまうと困る。

繰り返しのあるシミュレーション問題

AtCoder

- 問題のパターン
  - ○○を何回しなさい。
  - ○○を××するまで繰り返しなさい。
  - などなど。
- 本質的な部分はそんなに変わらない。
  - 繰り返すだけ！
  - forループや、whileループが使えれば、そんなに怖くない？

繰り返しのあるシミュレーション問題　実践

AtCoder

- 迷子のCDケース
  - http://arc007.contest.atcoder.jp/tasks/arc007_2

- 問題概要
- CDがN + 1枚存在する
  - CDケースはN枚しか存在しない
- CDを入れ替えるたびに、ケースに入っているCDの中身が入れ替わる。
- 最終的なCDの並び順を出力しなさい。

繰り返しのあるシミュレーション問題　実践

AtCoder

- やるべきこと
  - まずは、CDの枚数と、CDの入れ替え回数の読み取り
    - いつも通り標準入力で
  - 次に、CDの入れ替え処理のループ
    - 次に聞くCDを標準入力から受け取る
    - CDの入れ替えをシミュレーションする
      - まず、目的のCDを探す。
      - 次に、目的のCDと前聞いていたCDを入れ替える。
    - これをM回繰り返す
  - 最後に、CDケースに入っているCDの番号を出力する

繰り返しのあるシミュレーション問題　実践

AtCoder

- ソースコード
  - http://arc007.contest.atcoder.jp/submissions/140819

```
1  import java.util.Scanner;
2
3  public class Main{
4      public static void main(String[] args){
5          new Main().run();
6      }
7
8      void run()
9      {
10         Scanner cin = new Scanner(System.in);
11
12         int n = cin.nextInt();
13         int m = cin.nextInt();
14
15         //事前準備 各CDケースに、どのCDが入っているかを格納します。
16         //今CDプレイヤーに入っているCDは0なので、それを入れる。
17         int nowPlaying = 0;
18
19         //各CDケースに、入っているCDの番号を入れる
20         int[] CDCase = new int[n];
21         for(int i=0;i<n;i++) CDCase[i] = i + 1;
```

繰り返しのあるシミュレーション問題　実践

AtCoder

- ソースコード

```
        //M回の処理が必要となるので、繰り返し。
        for(int i=0;i<m;i++){
            //目的のCDを
            int targetCD = cin.nextInt();
            //目的のCDを探す
            int targetCase = -1;
            for(int j=0;j<n;j++){
                if(CDCase[j] == targetCD){
                    targetCase = j; break;
                }
            }
            //もし前聞いたCDと同じCDだったら、何もしない
            if(targetCase == -1) continue;
            //入れ替え処理を行う
            CDCase[targetCase] = nowPlaying;
            nowPlaying = targetCD;
        }

        //答えを出力します。
        for(int i=0;i<n;i++){
            System.out.println(CDCase[i]);
        }
    }
}
```

繰り返しのあるシミュレーション問題　実践

AtCoder

- こうした問題の注意点
  - 確かに、「書かれていることをそのまま実装すれば良いだけ」
    - だが、その具体的な実装方法が書かれているわけではない。
  - 例えば、今回の場合は、「入れ替える」という処理
    - 入れ替えるという動作が、以下の4ステップを省略している。
      - 「CDケースからCDを出す」（CD番号をメモしておく）
      - 「目的のCDが入っているケースを探す」(ケース番号をメモしておく)
      - 「目的のCDをCDプレイヤーに入れる」(再生中CD番号を更新する)
      - 「出したCDをCDケースに戻す」(メモしたケース番号のところに、メモしたCD番号を入れる)

繰り返しのあるシミュレーション問題　演習

AtCoder

- ARC011 鉛筆リサイクル工場の新技術
  - http://arc011.contest.atcoder.jp/tasks/arc011_1

- 問題概要
  - N本の鉛筆が製造済み。これを販売する。
  - m本の使用済み鉛筆から、新たにn本の鉛筆を製造することが出来る。
  - 製造された鉛筆は、全て使用され、回収できるものとする。
  - 再利用以外の製造を行わないとき、最終的に販売される鉛筆の本数を出力しなさい。

繰り返しのあるシミュレーション問題　演習

AtCoder

- 早くできた人は、この問題も解いてみよう！
  - 軽く触れますが、解説をするかはわかりません。

- ARC009 B問題 おとぎの国の高橋君
  - http://arc009.contest.atcoder.jp/tasks/arc009_2
- ARC006 B問題 あみだくじ
  - http://arc006.contest.atcoder.jp/tasks/arc006_2
- ABC004 C問題 入れ替え
  - http://abc004.contest.atcoder.jp/tasks/abc004_3

繰り返しのあるシミュレーション問題　演習

AtCoder

- 問題の解き方
  - 入力処理
  - ループ処理
    - 製造した鉛筆は全て使用されたことにする
    - 使用済み鉛筆から、さらに鉛筆が作れるなら、鉛筆を追加する。
      - もう作れないならループを抜ける
  - 出力処理

繰り返しのあるシミュレーション問題　演習

AtCoder

- ソースコード
  - http://arc011.contest.atcoder.jp/submissions/140836
  - サンプルの説明に合わせて解いた例

```
1  import java.util.Scanner;
2
3  public class Main{
4      public static void main(String[] args){
5          new Main().run();
6      }
7
8      void run()
9      {
10         Scanner cin = new Scanner(System.in);
11
12         int m = cin.nextInt();
13         int n = cin.nextInt();
14         int N = cin.nextInt();
15
16         //使われて残っている短い鉛筆の個数
17         int usedPencil = N;
18         //最終的に販売された鉛筆の個数
19         int ret = usedPencil;
20
```

繰り返しのあるシミュレーション問題　演習

AtCoder

- ソースコード

```
21          //ここからループ処理
22          while(true){
23              //再利用鉛筆が作れるかどうかを判定する
24              if(usedPencil >= m){
25                  //作れるのであれば、何セット作れるか判定する
26                  int numRecycle = usedPencil / m;
27
28                  //作成した分、残った鉛筆と販売数を増減させる。
29                  usedPencil -= numRecycle * m;
30                  usedPencil += numRecycle * n;
31                  ret += numRecycle * n;
32              }
33              //作れないのであればループから抜ける
34              else break;
35          }
36
37          System.out.println(ret);
38      }
39  }
40
```

繰り返しのあるシミュレーション問題　演習

AtCoder

- ソースコード
  - http://arc011.contest.atcoder.jp/submissions/140838
  - 手を抜いた例

```
21      //ここからループ処理
22      while(true){
23          //再利用鉛筆が作れるかどうかを判定する
24          if(usedPencil >= m){
25              //1セット分だけ、残った鉛筆と販売数を増減させる。
26              usedPencil -= m;
27              usedPencil += n;
28              ret += n;
29          }
30          //作れないのであればループから抜ける
31          else break;
32      }
33
34      System.out.println(ret);
35  }
36  }
```

繰り返しのあるシミュレーション問題　おまけ

AtCoder

- ARC009 B問題 おとぎの国の高橋君
  - http://arc009.contest.atcoder.jp/tasks/arc009_2

- 問題概要
  - ある国では、10進数が使われていて、同じ数字の記号が使われていますが、0以外の優先順位が違います。
    - 例えば、通常は0<1<2<3<4<5<6<7<8<9
    - この国では、0<5<2<4<6<3<9<8<7<1だったりする
  - 数字が複数与えられるので、この国で大きい順に並び替えなさい。

繰り返しのあるシミュレーション問題　おまけ

AtCoder

- 入力例

入力例 2

```
0 9 8 7 6 5 4 3 2 1
3
13467932
98738462
74392
```

出力例 2

```
74392
98738462
13467932
```

- 5 桁の数は 8 桁の数よりも小さいので、1 番は 74392 になります。
- 98738462 と 13467932 では最上位の 9 は1 より小さいので、98738462 が 2 番目、13467932 が 3 番目になります。

繰り返しのあるシミュレーション問題　おまけ

AtCoder

- やりかた
  - 文字列を順番に置換する
    - ０９８７６５４３２１だったら、
      - ０を０に置換する
      - ９を１に置換する
      - ・・・
      - のように、まず数字を置換してしまう。
  - その後、数字をソートする
    - 標準の関数でも使って貰えれば問題ないです。
  - 最後に、ソートする前の文字列を順番に出力する
    - 変換前と変換後の数字を持たせたTupleクラスを作るのが王道
    - HashMapなどで数字に対応する文字列をメモしても良い
    - 置換後の数字を100倍して元のindexの数字を突っ込む、なんて邪道技も競技プログラミング的にはナシではない

繰り返しのあるシミュレーション問題　おまけ

AtCoder

- やりかた　その他
  - 比較関数を自作しちゃってソート
  - 今回のテーマとあんまり合わなさそうなので扱いません。
    - ソート関数をデフォルトで用意されているものと仮定すると、「繰り返しのないシミュレーション問題」っぽくなる。
      - 内部的には何度も呼び出されてます。

繰り返しのあるシミュレーション問題　おまけ

AtCoder

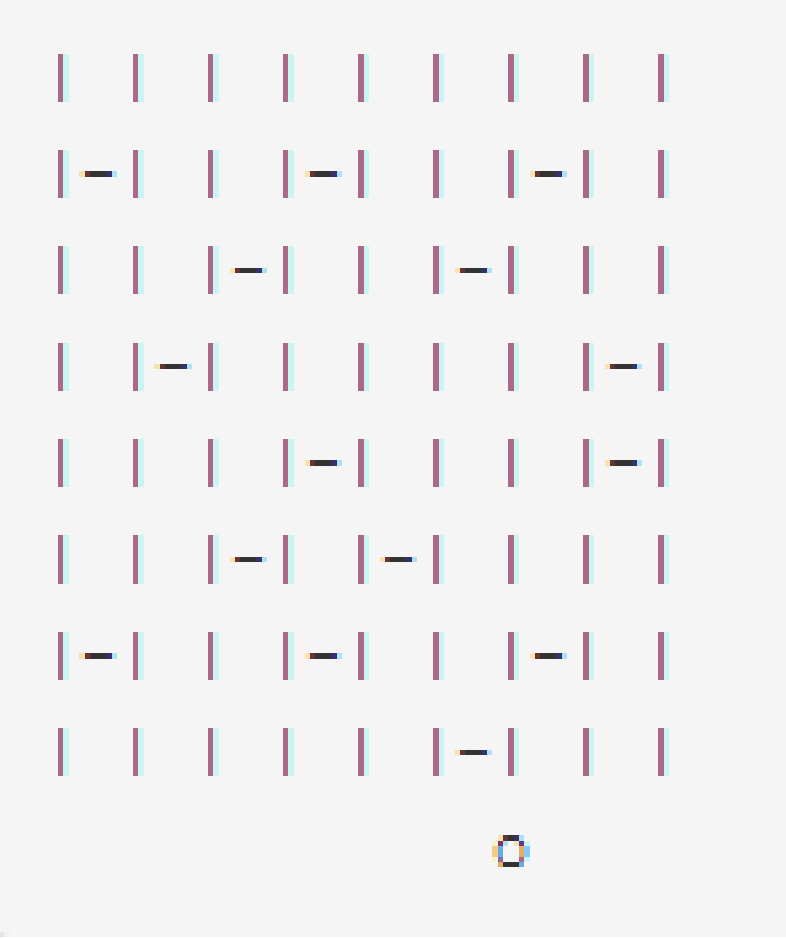
- ARC006 B問題 あみだくじ
  - http://arc006.contest.atcoder.jp/tasks/arc006_2

- 問題概要
  - あみだくじが与えられる
  - 当たり位置に辿り着くためにどれを選べば良いか答えなさい。

繰り返しのあるシミュレーション問題　おまけ

AtCoder

- こんなの



繰り返しのあるシミュレーション問題　おまけ

AtCoder

- やり方

ー気合で解く！

  - 実装が大変です。

繰り返しのあるシミュレーション問題　おまけ

AtCoder

- やり方
  - 入力を全部読み込み、一番下のoを探す
  - 当たり位置を、「そのまま」「1つ左」「1つ右」の、どれか適切な位置に動かす操作を、1行ずつ行う。
  - 最終的な当たり位置を出力

繰り返しのあるシミュレーション問題　おまけ

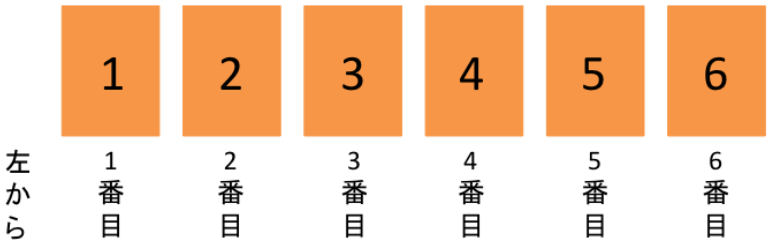
AtCoder

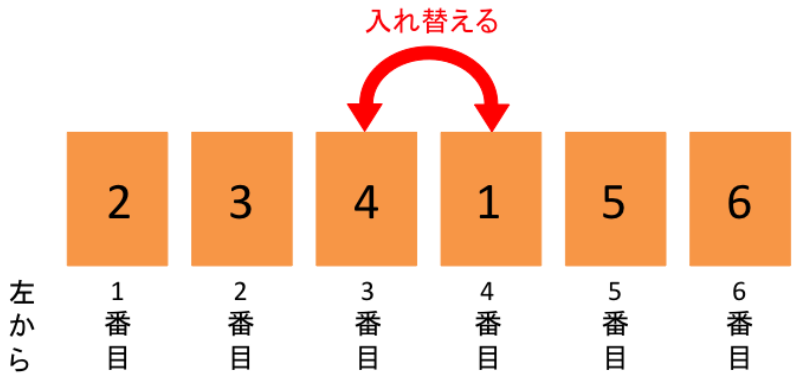
- ABC004 C問題 入れ替え
  - http://abc004.contest.atcoder.jp/tasks/abc004_3

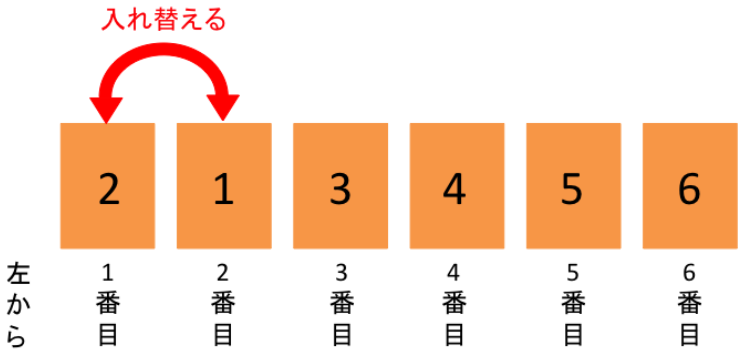
- 問題概要
  - 指定された順番でカードを入れ替える
  - n回入れ替えた時に、どのようなカードの並びになるか？
    - Nは1,000,000,000回まで！　←多い！

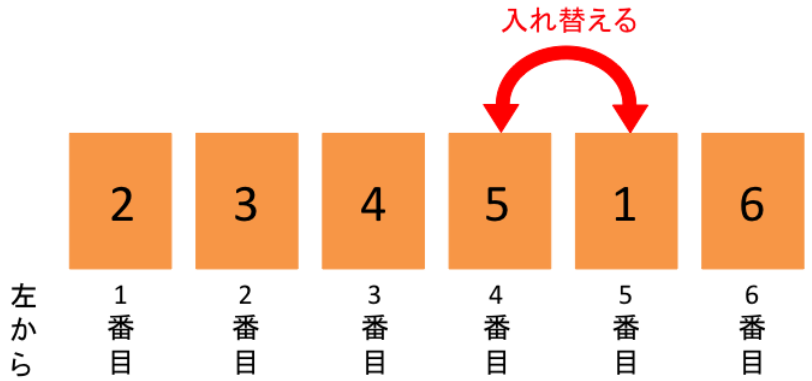
繰り返しのあるシミュレーション問題　おまけ

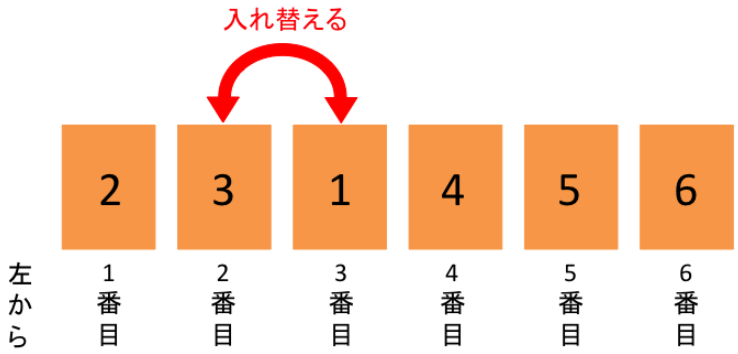
AtCoder

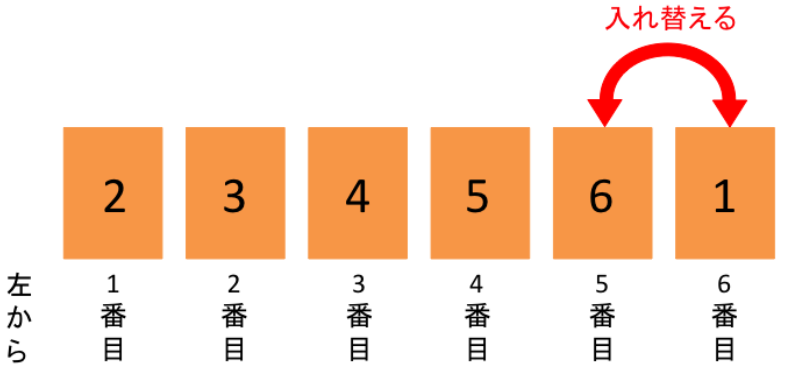












繰り返しのあるシミュレーション問題　おまけ

AtCoder

- やり方
  - カードを指定された回数だけ入れ替える
    - ・・・・だけじゃ実はダメ
    - 回数が多過ぎる！
  - 途中でループするので、ループする分を省略して、残った分だけ入れ替える
    - 30回でループします。
  - この辺りの、計算時間がかかりすぎてしまうパターンは、第3回以降にしっかり勉強します！

繰り返しのあるシミュレーション問題　演習

AtCoder

- このジャンルで難しい問題のパターン
  - 実装が複雑
    - やるべきことが多かったり、混乱してしまうパターン
    - 何をやっていいか解らなくなってしまうことも多々
    - 紙に書くなどして、一度しっかり整理してから実装しよう
  - 繰り返す回数が異様に多い
    - 計算時間が長くなってしまう。
    - このパターンは第３回以降で勉強します！

繰り返しのあるシミュレーション問題　攻略方法！

AtCoder

- 「思い通りの操作が出来る」ことが一番大切！
  - 「こう動かしたい」と思ったのが実装できれば、そう苦労する問題ではないはず。
  - 繰り返しのないパターンと一緒。ただこちらの形式の方が、少し複雑にしやすい。
- 計算時間が増えてしまうパターンは後回し
  - 計算時間を短くする話は、

AtCoder

休憩！

次の開始時刻は15:40から

全探索問題

1. 全探索問題って？
2. 色々な全探索
3. 一番基本的な全探索

全探索問題って？

AtCoder

- 全部のパターンをしらみつぶしに調べる手法
  - ○○なパターンが何通りあるか答えなさい。
    - 全パターン調べて、条件を満たすパターンをカウントする
  - ○○なパターンが存在するか調べなさい。
    - 全パターン調べて、条件を満たすパターンが存在するならYes、存在しないならNoを出力
  - ○○するとき、最大値を求めなさい。(or最小値)
    - 全パターンを調べて、一番大きいもの・一番小さいものを求める
- とにかく、全パターンを列挙することが重要！

色々な全探索

AtCoder

- 全パターンの列挙って簡単？
  - 問題によって全然違う！

- 問題の例に対して、どのような全列挙を行うべきか、考えてみましょう。
  - 1から100000までの数字の中に、素数がいくつあるか答えなさい。
  - アルファベット3文字で構成された文字列のうち、画数が5以下で書けるものが何通りあるか求めなさい。
  - 将棋の駒の香車があります。1マス目から9マス目まで移動する時、2つ飛ばしで移動する動きがないパターンの数を答えなさい。
  - 3*3のパネルに、9種類の数字を1つずつ置くことが出来ます。魔法陣が作れるかどうかを判定しなさい。
  - 10問の○×クイズを連続して解きます。6問以上正解して、なおかつ3問連続正解を含む、解答の仕方が何通りあるか答えなさい。
  - 10個のりんごを、3人で分けます。分け方が何通りあるかを答えなさい。

色々な全探索

AtCoder

- 問題例1
- 1から100000までの数字の中に、素数がいくつあるか答えなさい。

- パターンの全列挙

色々な全探索

AtCoder

- 問題例1
- 1から100000までの数字の中に、素数がいくつあるか答えなさい。

- パターンの全列挙
  - 数字のパターンの全列挙をする
  - 1,2,3,4,5,6,……

全探索問題って？

AtCoder

- 問題例2
- アルファベット3文字で構成された文字列のうち、画数が5以下で書けるものが何通りあるか求めなさい。

- パターンの全列挙

色々な全探索

AtCoder

- 問題例2
- アルファベット3文字で構成された文字列のうち、画数が5以下で書けるものが何通りあるか求めなさい。

- パターンの全列挙
  - アルファベットのパターンを全部並べる
    - AAA, AAB, AAC, AAD, AAE, AAF,…

色々な全探索

AtCoder

- 問題例3
  - 将棋の駒の香車があります。1マス目から9マス目まで移動する時、2つ飛ばしで移動する動きがないパターンの数を答えなさい。

- パターンの列挙

色々な全探索

AtCoder

- 問題例3
  - 将棋の駒の香車があります。1マス目から9マス目まで移動する時、2つ飛ばしで移動する動きがないパターンの数を答えなさい。

- パターンの列挙
  - 移動方法を全部列挙すれば良い
    - 1 → 9
    - 1 → 8 → 9
    - ….
    - 1 → 2 → 3 → 4 → 5 → 6 → 7 → 8 → 9

色々な全探索

AtCoder

- 問題例3
  - 将棋の駒の香車があります。1マス目から9マス目まで移動する時、2つ飛ばしで移動する動きがないパターンの数を答えなさい。

- パターンの列挙　その2
  - 移動した距離の全列挙、みたいな変則的なのもある
    - 8
    - 7 → 1
    - ….
    - 1 → 1 → 1 → 1 → 1 → 1 → 1 → 1 → 1

色々な全探索

AtCoder

- 問題例4
- 3*3のパネルに、9種類の数字を1つずつ置くことが出来ます。魔法陣が作れるかどうかを判定しなさい。
  - 仮に、使える数字を1,2,3,4,5,6,7,8,9の9個とする。

- パターンの全列挙

色々な全探索

AtCoder

- 問題例4
- 3*3のパネルに、9種類の数字を1つずつ置くことが出来ます。魔法陣が作れるかどうかを判定しなさい。
  - 仮に、使える数字を1,2,3,4,5,6,7,8,9の9個とする。

- パターンの全列挙
  - 3*3の盤面を、全通り列挙すれば良い
    - ２次元配列か何かになるのかな？

1	2	3
4	5	6
7	8	9

1	2	3
4	5	6
7	9	8

1	2	3
4	5	6
8	7	9

……

9	8	7
6	5	4
3	2	1

色々な全探索

AtCoder

- 問題例5
  - 10問の〇×クイズを連続して解きます。6問以上正解して、なおかつ3問連続正解を含む、解答の仕方が何通りあるか答えなさい。
    - 解答は、仮にoxoxoxoxoxだったとします。
- パターンの全列挙

色々な全探索

AtCoder

- 問題例5
  - 10問の〇×クイズを連続して解きます。6問以上正解して、なおかつ3問連続正解を含む、解答の仕方が何通りあるか答えなさい。
    - 正解は、仮に1問目から順番にo,x,o,x,o,x,o,x,o,xだったとします。

- パターンの全列挙
  - 10問に対して、〇×のどちらを答えたかを列挙する
    - o, o, o, o, o, o, o, o, o ,o
    - o, o, o, o, o, o, o, o, o, x
    - ….
    - x, x, x, x, x, x, x, x, x, x

色々な全探索

AtCoder

- 問題例5
  - 10問の〇×クイズを連続して解きます。6問以上正解して、なおかつ3問連続正解を含む、解答の仕方が何通りあるか答えなさい。
    - 正解は、仮に1問目から順番にo,x,o,x,o,x,o,x,o,xだったとします。

- パターンの全列挙2
  - 10問に対して、正解・不正解を列挙する
    - o, o, o, o, o, o, o, o, o ,o
    - o, o, o, o, o, o, o, o, o, x
    - ….
    - x, x, x, x, x, x, x, x, x, x

色々な全探索

AtCoder

- 問題例6
  - 10個のりんごを、3人で分けます。分け方が何通りあるか答えなさい。

- 全列挙のパターン

色々な全探索

AtCoder

- 問題例6
  - 10個のりんごを、3人で分けます。分け方が何通りあるか答えなさい。

- 全列挙のパターン
  - 10こ、0こ、0こ
  - 9こ、1こ、0こ
  - …
  - 3こ、3こ、4こ
  - …
  - 0こ、0こ、10こ

色々な全探索

AtCoder

- あらためて、どれが簡単かを考えてみよう
  - 1から100000までの数字の中に、素数がいくつあるか答えなさい。
  - アルファベット3文字で構成された文字列のうち、画数が5以下で書けるものが何通りあるか求めなさい。
  - 将棋の駒の香車があります。1マス目から9マス目まで移動する時、2つ飛ばしで移動する動きがないパターンの数を答えなさい。
  - 3*3のパネルに、9種類の数字を1つずつ置くことが出来ます。魔法陣が作れるかどうかを判定しなさい。
  - 10問の○×クイズを連続して解きます。6問以上正解して、なおかつ3問連続正解を含む、解答の仕方が何通りあるか答えなさい。
  - 10個のりんごを、3人で分けます。分け方が何通りあるかを答えなさい。

色々な全探索

AtCoder

- かんたん
  - 1から100000までの数字の中に、素数がいくつあるか答えなさい。
  - アルファベット3文字で構成された文字列のうち、画数が5以下で書けるものが何通りあるか求めなさい。
  - 10個のりんごを、3人で分けます。分け方が何通りあるかを答えなさい。
- むずかしい
  - 将棋の駒の香車があります。1マス目から9マス目まで移動する時、2つ飛ばしで移動する動きがないパターンの数を答えなさい。
  - 3*3のパネルに、9種類の数字を1つずつ置くことが出来ます。魔法陣が作れるかどうかを判定しなさい。
  - 10問の○×クイズを連続して解きます。6問以上正解して、なおかつ3問連続正解を含む、解答の仕方が何通りあるか答えなさい。
- この差は何だろう？

色々な全探索

AtCoder

- 素数の数を求める問題
  - 1から100000まで、forループで列挙すれば良い

– アルファベット3文字について、画数を考える問題
  - まず1文字目を、AからZまでforループで決定する
  - 次に2文字目を、AからZまでforループで決定する
  - 最後に3文字目を、AからZまでforループで決定する
  - 以上のような、3重ループで、全部の文字列を試すことが出来る。
    – やったね！

色々な全探索

AtCoder

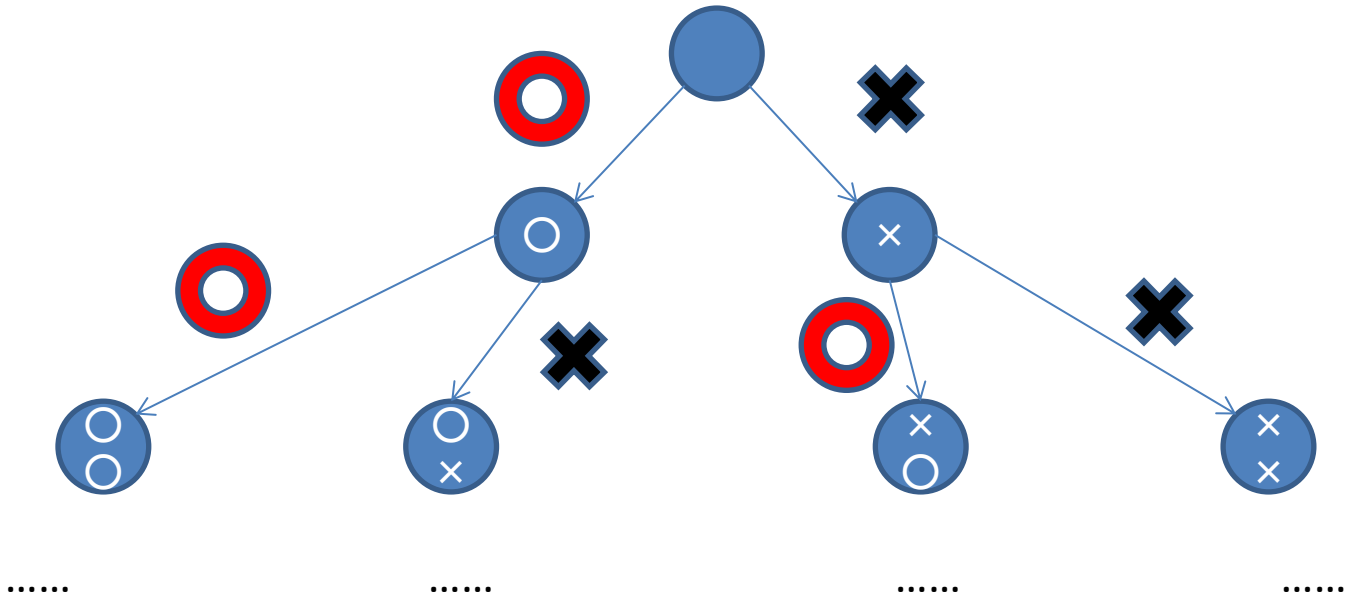
- 10個のりんごを、3人で分けます。分け方が何通りあるかを答えなさい。
  - まず一人目が貰うりんごの数を、forループで列挙(0個から10個の間)
  - その後、二人目が貰うりんごの数を、forループで列挙
    - 今度は、0個から10個ではなく、0から(10-一人目のりんごの数)の間
  - 三人目に貰うりんごの数は、引き算で自動的に決まる。
  - 2つのforループで、全部のパターンが列挙出来る！
    - やったね！

- forループで全列挙が簡単に出来る！

色々な全探索

AtCoder

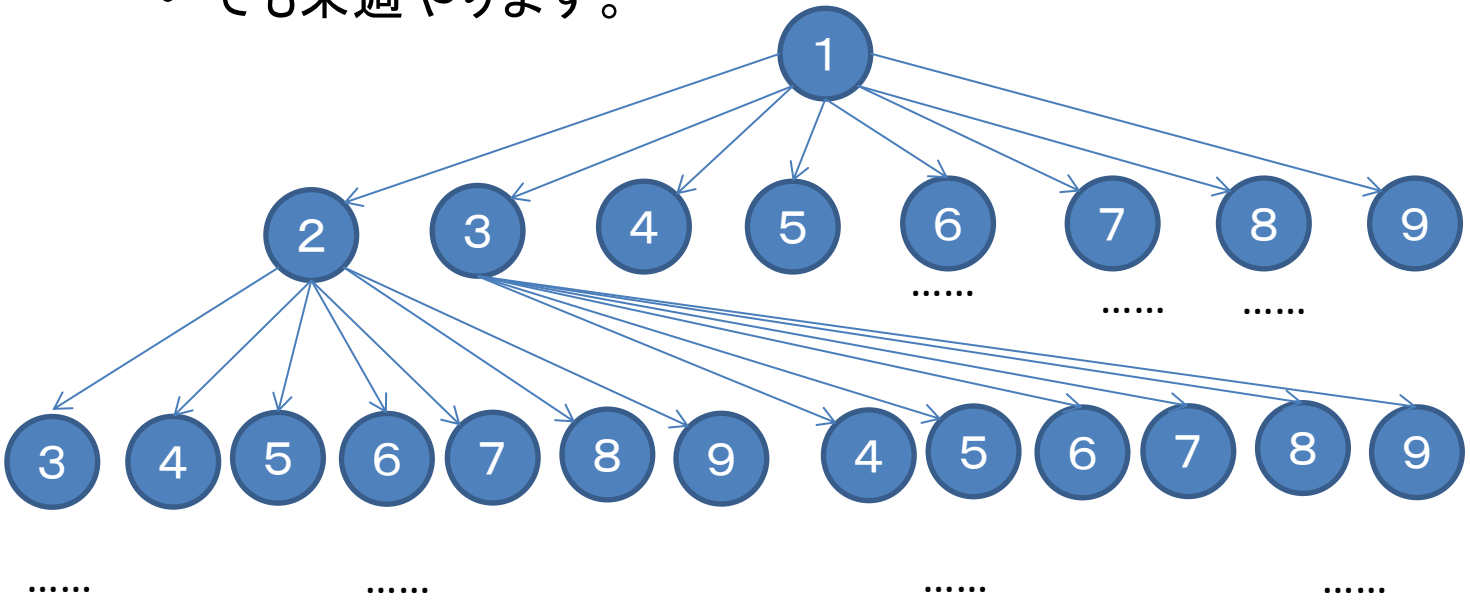
- 他の問題は、全列挙がちょっと難しい
- ○×クイズの問題



色々な全探索

AtCoder

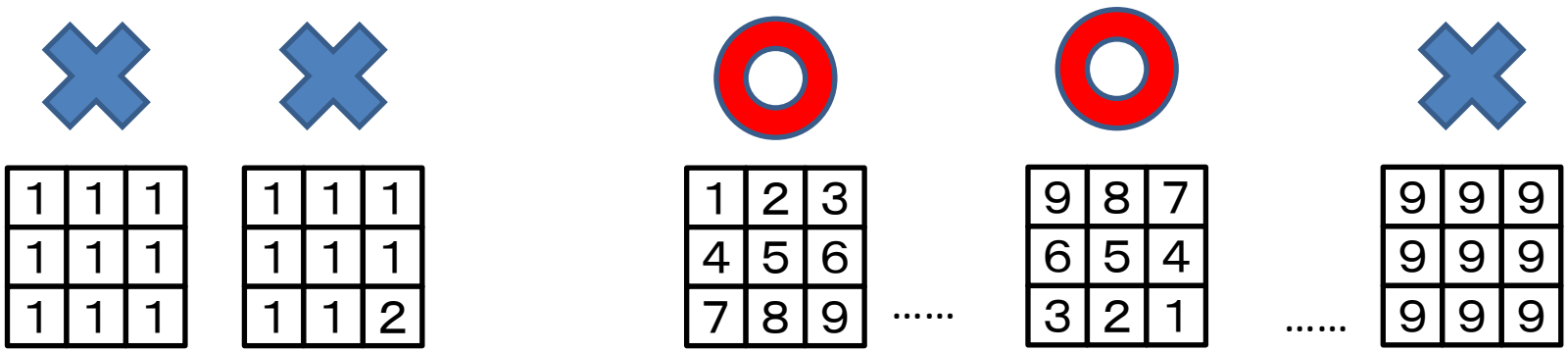
- 他の問題は、全列挙がちょっと難しい
- 香車の問題
  - こんなん実装出来るか！って感じになる。
    - でも来週やります。



色々な全探索

AtCoder

- 3*3のパネルに、9種類の数字を1つずつ置くことが出来ます。魔法陣が作れるかどうかを判定しなさい。
  - 実はこれもforループで列挙可能
    - 9回forループを回して、被った数字が出来てしまったら、またやり直す。
    - 全ての数字がバラバラになったものだけ、パターンとして数え上げれば良い。



色々な全探索

AtCoder

- 他の2問もforループで解ける？
  - 香車の問題
    - 2,3,4,5,6,7,8マス目に対して、「止まる」「止まらない」を判定出来れば、経路が決まる。
    - つまり、7つの、0,1のみのforループで表現できる！！
  - ○×の問題
    - 1～10問目に対して、「○と答えた」「×と答えた」を0,1で表現してあげれば、全てのパターンが列挙出来る
    - つまり、10つの、0,1のみのforループで表現できる！！
- でも、こんな実装はしちゃだめです！
  - 2日目に説明するので、それまで我慢してね！

色々な全探索

AtCoder

- 先ほどの実装が良くない理由
  - forループが多過ぎる！
    - あんまり多いと訳が分からなくなります。
  - そもそもforループ数が固定じゃないと上手くいきません
    - N問の○×クイズが……みたいな問題に対応できない。
  - 単純なforループだけで書けるプログラムはあんまり多くない。

色々な全探索

AtCoder

- 簡単にforループで全列挙が出来る条件は？
  - 分岐の回数がある程度少なく、回数が決まっている
    - N個のリンゴをM人で分けます。とかは無理！
  - ループごとに、選択肢が有限個で決まっている
    - 整数なら良いけど、0以上1以下の実数、とかは無理！
  - この2つが満たされていれば、全列挙はforループで簡単に書ける！
    - 今日は、このforループで書ける全列挙をマスターしよう！

一番基本的な全探索　実践

AtCoder

- 天下一プログラマーコンテスト2012 予選B
- A問題　孫子算経
  - http://tenka1-2012-qualb.contest.atcoder.jp/tasks/tenka1_2012_5

- 問題概要
  - いくつか物が存在する。
  - その物の個数は、1～127の範囲である。
  - その個数を3で割るとaあまり、5で割るとbあまり、7で割るとc余る。
  - 個数としてあり得る数を列挙しなさい。

一番基本的な全探索　実践

AtCoder

- やりかた
  - 1から127の全ての数字に対してforループを回す。
    - それぞれの条件を満たすかどうか調べる。
    - もし満たすのであれば出力をする。

一番基本的な全探索　実践

AtCoder

- ソースコード
  - http://tenka1-2012-qualb.contest.atcoder.jp/submissions/140848

```
1  import java.util.Scanner;
2
3  public class Main{
4      public static void main(String[] args){
5          new Main().run();
6      }
7
8      void run()
9      {
10         Scanner cin = new Scanner(System.in);
11         //入力
12         int a = cin.nextInt();
13         int b = cin.nextInt();
14         int c = cin.nextInt();
15
16         //全ての数字について試す
17         for(int i=1;i<=127;i++){
18             //条件を満たしていれば出力
19             if(i % 3 == a && i % 5 == b && i % 7 == c)
20                 System.out.println(i);
21         }
22
23     }
24 }
25
```

一番基本的な全探索　演習

AtCoder

- ARC001 A問題 センター採点
  - http://arc001.contest.atcoder.jp/tasks/arc001_1

- 問題概要
  - N問の4択問題と、その正解が与えられます。
  - 高橋君は、全てのマークを同じにしたことは覚えていますが、何につけたかは覚えていません。
  - 高橋君の取り得る最高点、最低点を出力しなさい。

一番基本的な全探索　演習

AtCoder

- 出来た人は次の問題にチャレンジ！
  - ARC004 A問題 2点間距離の最大値
    - http://arc004.contest.atcoder.jp/tasks/arc004_1
  - ARC018 B問題 格子点と整数
    - http://arc018.contest.atcoder.jp/tasks/arc018_2
    - こっちは多分解説しません。
  - ABC004 D問題 マーブル
    - http://abc004.contest.atcoder.jp/tasks/abc004_4
    - 難問です。多分こっちも解説しません。

一番基本的な全探索　演習

AtCoder

- やりかた
  - 入力
  - 選択肢の全てを列挙（選択肢の1,2,3,4の4パターン）
    - その選択肢を選んだ時に、何点取れているか調べる。
    - 最低点、および最高点を更新する。
  - 出力

一番基本的な全探索　演習

AtCoder

- ソースコード
  - http://arc001.contest.atcoder.jp/submissions/140849

```
1  import java.util.Scanner;
2
3  public class Main{
4      public static void main(String[] args){
5          new Main().run();
6      }
7
8      void run()
9      {
10         Scanner cin = new Scanner(System.in);
11         //入力
12         int N = cin.nextInt();
13         String st = cin.next();
14
15         //最小値、最大値を格納する
16         int retmax = 0;
17         int retmin = N;
18
```

一番基本的な全探索　演習

AtCoder

- ソースコード

```
19          //全部のパターン(1,2,3,4)について調べる
20          for(int i=1;i<=4;i++){
21              int count = 0;
22              //数字の1と文字の'1'は違うので、変換を行う。
23              char now = (char)('0' + i);
24
25              //全部の問題に対して、何問正解か調べる。
26              for(int j=0;j<N;j++){
27                  if(st.charAt(j) == now) count++;
28              }
29              //最小値、最大値を更新する
30              retmax = Math.max(retmax, count);
31              retmin = Math.min(retmin, count);
32          }
33          //出力
34          System.out.println(retmax + " " + retmin);
35      }
36  }
37
```

一番基本的な全探索　演習

AtCoder

- ARC004 A問題 2点間距離の最大値
  - http://arc004.contest.atcoder.jp/tasks/arc004_1

- 問題概要
  - 点がN個与えられます。
  - 点と点の距離のうち、最も遠いものを出力しなさい。

一番基本的な全探索　演習

AtCoder

- 出来た人は次の問題にチャレンジ！
  - ARC018 B問題 格子点と整数
    - http://arc018.contest.atcoder.jp/tasks/arc018_2
    - こっちは多分解説しません。
  - ABC004 D問題 マーブル
    - http://abc004.contest.atcoder.jp/tasks/abc004_4
    - 難問です。多分こっちも解説しません。

一番基本的な全探索　演習

AtCoder

- やりかた
  - まず、1つ目の点を、全部の点から1つ選ぶ
  - 次に、2つ目の点を、全部の点から1つ選ぶ
    - この2つのループで、全列挙が出来ているのがポイント！
  - あとは、それぞれの点同士の距離を求めるだけ。
  - 参考
    - 点同士のユークリッド距離
    - L = √((x1 – x2)^2 + (y1 – y2)^2)

一番基本的な全探索　演習

AtCoder

- ソースコード
  - http://arc004.contest.atcoder.jp/submissions/140850

```
8    void run()
9    {
10       Scanner cin = new Scanner(System.in);
11       //入力
12       int N = cin.nextInt();
13       int[] x = new int[N];
14       int[] y = new int[N];
15       for(int i=0;i<N;i++){
16          x[i] = cin.nextInt();
17          y[i] = cin.nextInt();
18       }
19
20       double maxDist = 0;
21       //1つ目の点を全通り試す
22       for(int i=0;i<N;i++){
23          //2つ目の点を全通り試す。iより小さい点は調べる必要がない
24          for(int j=i+1;j<N;j++){
25             maxDist = Math.max(maxDist, getDist(x[i],y[i],x[j],y[j]));
26          }
27       }
28       System.out.println(maxDist);
29   }
30
31   //距離を調べる関数
32   double getDist(double x1, double y1, double x2, double y2){
33       return Math.sqrt((x1-x2)*(x1-x2) + (y1-y2)*(y1-y2));
34   }
```

一番基本的な全探索　攻略方法！

AtCoder

- forループを使った全探索は、積極的に使おう！
  - これで書けるなら、実装量は非常に少ないことが多い！
  - 列挙出来てしまえば後は簡単。
  - まずは、この方法で解けないかを調べてみるべし。

一番基本的な全探索　攻略方法！

AtCoder

- 問題が難しくなると・・・？
  - 全探索をするまでに、1ステップ思考が増えることが多い。
    - 「これは〇〇だから、この範囲だけ全探索すれば良い」みたいな
    - 「特定の範囲を全探索をすれば良い」が自明ではなく、それ自体に気付くことが難しい。
  - 列挙が出来ても、その後の処理が難しいパターンも。
    - 「格子点と整数」では、3つの点を列挙出来ても、その後三角形の面積を求める部分で苦戦した人が多数

本日のまとめ

本日のまとめ

AtCoder

- 競技プログラミングの雰囲気を掴む！
  - 問題の形式、提出の仕方などを覚える
- 簡単なシミュレーション問題を解けるように！
  - 問題文に書かれていることを、素直に実装する
- forループで出来る全探索の問題を解けるように！
  - 全てのパターンを列挙して、それぞれについて調べる