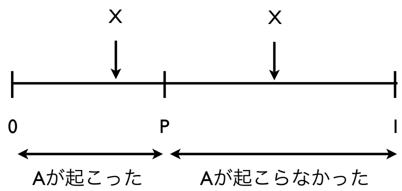
モンテカルロ法

- 乱数を用いてシミュレーションや数値計算を行う手法の総称
- 物理学や生物学のシミュレーションに良く用いられる
- 具体例：コイン投げ、ランダムウォーク（乱歩）、など、確率論的な事象の変化をアルゴリズムとして記述して実行する

確率的な事象のプログラム実装

- 確率 P で起こる事象 A をプログラムとして実装（$0 \le P \le 1$）
  - [0, 1) の疑似一様乱数 X を生成
  - X < P なら、事象 A が起こったと見なす。そうでなければ起こらなかったと見なす



X は区間 [0, 1) で一様（と期待する）

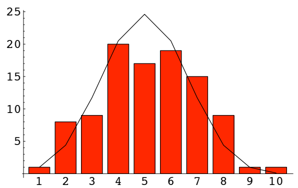
コイン投げ

- 正しく造られたコインは裏表がでる確率は P = 1/2 である。
- n 回コインを投げたとき、i 回表がでる確率（i = 0, 1, 2, ... n）は、二項分布で与えられる

$$P_n(i) = \ {}_n\mathrm{C}_i \left(\frac{1}{2}\right)^n$$

具体例：P = 1/2, n = 10 を 100 回繰り返したとき、表が出た回数

{7, 7, 9, 8, 4, 7, 8, 3, 7, 3, 1, 4, 2, 4, 5, 5,
6, 3, 8, 7, 5, 2, 6, 4, 4, 6, 3, 3, 5, 6, 2, 5, 8,
2, 6, 2, 6, 4, 6, 7, 10, 4, 5, 8, 5, 4, 4, 5, 4,
5, 3, 3, 8, 5, 5, 6, 5, 5, 4, 6, 6, 4, 2, 7, 2, 6,
6, 6, 4, 7, 5, 6, 3, 6, 7, 3, 6, 6, 8, 5, 6, 4, 7,
4, 2, 4, 5, 8, 7, 4, 4, 5, 7, 4, 4, 7, 7, 8, 6, 7}



問題 1

- コインを n 回投げる試行（ベルヌーイ試行）を k 回繰り返す。各試行で表が出た回数を i とする。 i をファイルに書き出せ。
- 試行数 k を十分大きくとったとき、i の分布図を描け。また理論分布と比較せよ。n の値は適当でよい。

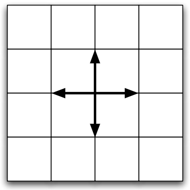
1回の試行で表が出た回数を戻り値として返す関数を定義

```
int binomial(int num, double prob)
{

  .....
  return count;
}
```

ランダムウォーク

- 2 次元格子空間を考える。各個体は格子上に存在し、単位時間内に隣接する 4つの格子のいずれかへ等しい確率 1/4 で移動する。
- 初期分布として原点に N 個体存在する状態を考える。時刻 t での個体の空間分布はどのようなものか？
- また、時刻 t と原点から最も離れた個体の距離との関係は？

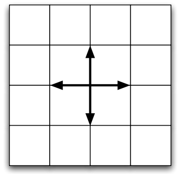


問題 2

- N 個体の位置を整数型の 2次元配列で表現するランダムウォークに取り組む
- 初期状態として全ての個体は原点に位置するとする
- 以下を繰り返す（時刻のループ）
  - 次のアルゴリズムに従って、N 個体の座標を変化させる

各個体の x, y 座標は

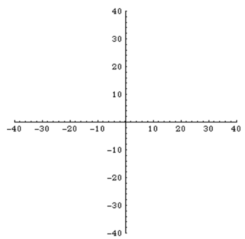
確率 1/4 で x++
確率 1/4 で x--
確率 1/4 で y++
確率 1/4 で y--



2次元格子上の乱歩

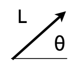
```
data = ReadList[ "data", Real, RecordLists -> True ];
data2 = Map[ Partition[#, 2] &, data];
Length[data2]
{ Max[data2], Min[data2] }

Do[
  ListPlot[ data2[[i]], PlotRange -> {{-40, 40}, {-40, 40}}, AspectRatio ->1 ], {i, 1, Length[data2]}
]
```



問題3

- 個体の位置は格子上ではなく、連続空間上にあるとする（x, y 座標が実数）
- 単位時間後の個体の移動に関して適当なルールを設定し、N 個体の移動分散の様子を調べよ。ルールとしては下記が考えられる
  - ルール 1：一定の距離 L だけ移動するが、方角は任意の方向に等しい確率で移動
  - ルール 2：移動距離は一定 L だが、方向は N 個体の重心へ向かう
  - ルール 3：周囲の個体の数を数えて、混み合ってくると長移動距離 L' を行う



自分で設定したルールの下で、空間分布の時間変化および
時刻 t と原点から最も離れた個体の距離との関係を調べよ