HW2 PB6

**part A, Satisfiability Intro [easy].** A boolean formula is satisfiable if there exists some variable assignment that makes the formula evaluate to true. Namely, a boolean formula is satisfiable if there is some row of the truth table that comes out true. Determining whether an arbitrary boolean formula is satisfiable is called the *Satisfiability Problem*. There is no known efficient solution to this problem, in fact, an efficient solution would earn you a million dollar prize. While this is hard problem in computer science, not all instances of the problem are hard, in fact, determining satisfiability for some types of boolean formulae is easy.

$(A \Rightarrow B) \equiv B \vee \neg A$

$(\neg B \Rightarrow \neg A) \equiv$

i. First, let's consider why this would be hard. If you knew nothing about a given boolean formula other than that it had $n$ variables, how large is the truth table you would need to construct? Please indicate the number of columns and rows as a function of $n$

ii. Now consider the following 100 variable formula.

$\neg x_k \vee x_{k+1}$ → particular 2CNF

$x_1$

$$x_1 \wedge (\neg x_1 \vee x_2) \wedge (\neg x_2 \vee x_3) \wedge (\neg x_3 \vee x_4) \wedge \ldots \wedge (\neg x_{99} \vee x_{100})$$

$(x_1 \Rightarrow x_2) \wedge (x_2 \Rightarrow x_3) \wedge \ldots \wedge (x_{99} \Rightarrow x_{100})$

Without constructing a truth table, how many satisfying assignments does this formula have, explain your answer.

iii Now consider an arbitrary 3-DNF formula with 100 variables and 200 clauses. 3-DNF means that the formula is in disjunctive normal form and each clause has three literals. (A literal is the instantiation of the variable in the formula, so for $x$, $\neg x$ or $x$.) An example might be something like:

$$(\neg x_1 \wedge x_3 \wedge x_{10}) \vee (\neg x_3 \wedge x_{15} \wedge \neg x_{84}) \vee (x_{17} \wedge \neg x_{37} \wedge x_{48}) \vee \ldots \vee (\neg x_{87} \wedge \neg x_{95} \wedge x_{100})$$

What is the largest size truth table needed to solve this problem. What is the maximum number of such truth tables needed to determine satisfiabilty.

HON PB2: general 2-CNF Formula
find x bool assignment.

**part B: 2CNF-SAT [hard].** The 2CNF-SAT instance is a boolean CNF formula with 2 variables in each clause, "OR" inside clauses, "AND" between clauses. There are $m$ boolean variables $x_1, x_2, ..., x_m$) and $n$ clauses $C_1, C_2, ..., C_n$). Every variable and its negation appears in at least one clause. Such formula is given as input in format redundantly :
- for each variable there is a list of clauses containing it
- for each clause there there are 2 variables

2CNF general

For example the formula $(x_1 \vee \neg x_2) \wedge (x_2 \vee x_3) \wedge (\neg x_1 \vee x_3) \wedge (\neg x_2 \vee \neg x_3)$ will be given as:

$m = 3, n = 4$
$x_1 : C_1$
$\neg x_1 : C_3$
$x_2 : C_2$
$\neg x_2 : C_1, C_4$
$x_3 : C_2, C_3$
$\neg x_3 : C_4$
$C_1 : x_1, \neg x_2$
$C_2 : x_2, x_2$
$C_3 : \neg x_1, x_3$
$C_4 : \neg x_2, \neg x_3$

$x_2 \Rightarrow x_1$
$\neg x_1 \Rightarrow \neg x_2$

transform each clause into 2 implications

Want emphasize: procedure (recipe)

Your task is to design a strategy that determines, for a given formula, the boolean assignments for the variables such that all clauses are satisfied, thus the formula is true (if more such assignments are possible, you only need to output one). If no such assignment is possible, output "FALSE".

As established inpart A, there are $2^m$ possible assignments for the variable set. So if one were to build the truth table and "brute force" search all rows/assignments until one works, it would take exponential time — not good! Instead: do trial and error, but in a smart way that only tries at most $2 * m^2$ boolean assignments.

Your strategy can be pseudocode, or you can informally describe a procedure with bullets and English statements. You can write in your procedure statements like
* $x = x_1$

* foreach $C$ containing variable $x$ {
- - - -
}
* $C$= next clause, or $C$ = next clause containing $x$

* loop C through all clauses that contain $x$ or $\neg x$

* for each $x \in C$ {
- - - -
}
* $y$ = the other variable in clause $C$, other than $x$ or $\neg x$

# Lecture 8 COUNTING + Intro non-PB2

- Basic Definitions / recipies (regular)
  - PHP, set op, Sum Rule → HW 3+4
    Product Rule → rec4
    Comb / Perm
    Balls into Bins

- Advanced techniques
  ex. Binomial app / Th
  - Sum Rule generalized
  - Catalan #

- Adv thinking
  - creativity
  - modeling, complex pb

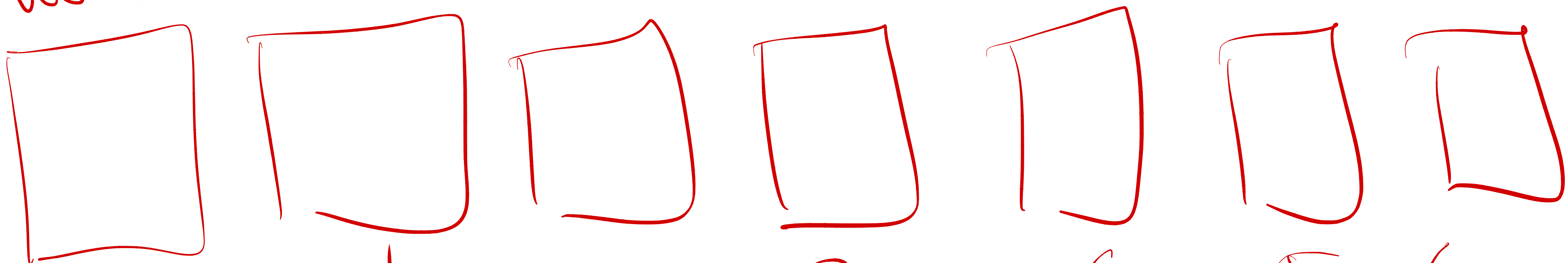
## $\boxed{\text{PHP 1}}$

100 integers $\Longrightarrow$ $(\exists)$ select 15 of them

any diff of 2 = multiple of 7

$a-b = 7k \iff 7 \mid a-b \iff \boxed{a \equiv b \bmod 7}$

mod 7 $\Longrightarrow$ 7 boxes



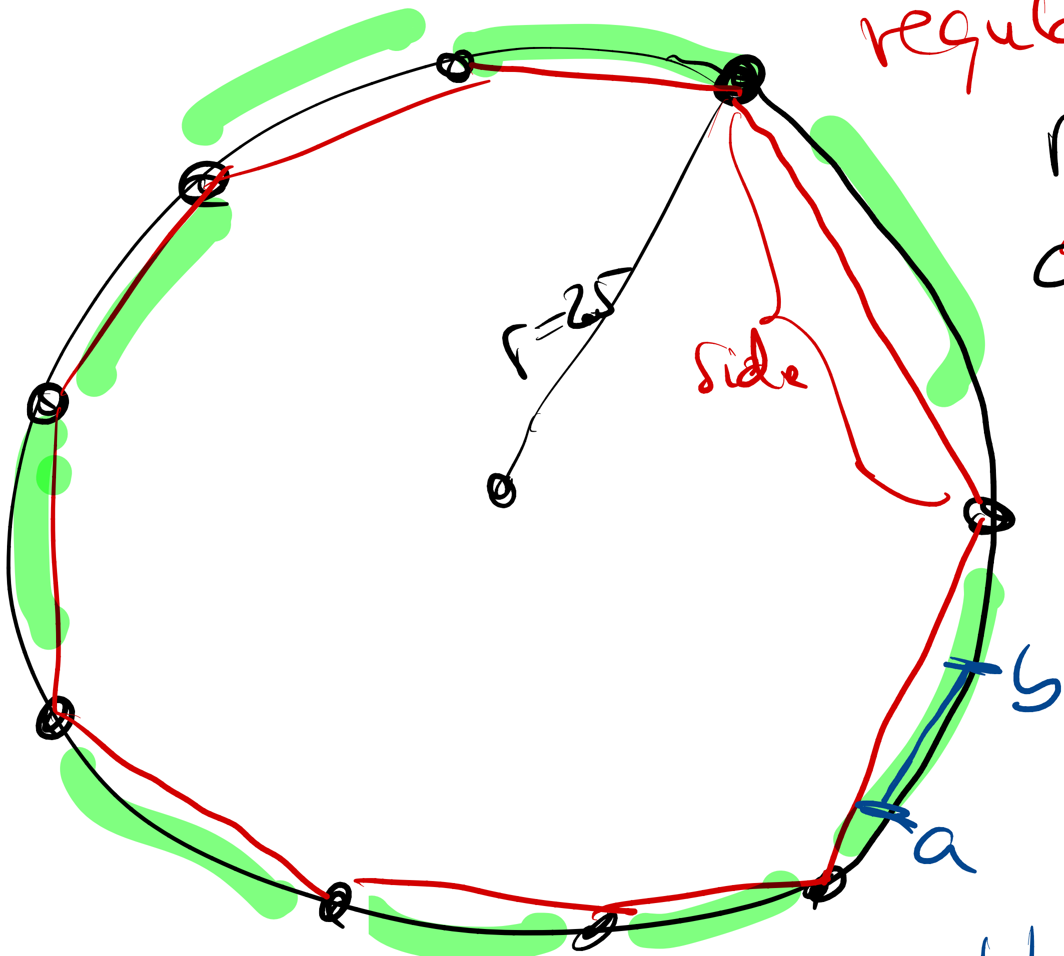
rem=0 1 2 3 4 5 6

100 integers
7 boxes } $\xrightarrow{\text{PHP}}$ $\exists$ box with $\geq$ 1 box has at least $\left\lceil \frac{100}{7} \right\rceil \geq 15$

same rem at 7 $\Longrightarrow$ $|a-b|$ = multiple of 7

10 point on a circle of diameter $= 2r = 5$

PHP2 $\Rightarrow \exists$ 2 of them at $\text{dist}(a,b) < 2$.
a, b




regular 9-gon (equal sides)

$r = 2.5 \Rightarrow$ side $\simeq 1.71$?

9-gon splits circle into 9 regions (green)

10 points on circle

$\Downarrow$ PHP

2 of them same region

geom: $|ab| <$ side 9-gon $\simeq 1.71$

**COUNT 1**

Two decks of cards are mixed together, total 104 cards where each card appears exactly twice. How many distinct permutations are there of all 104 cards?

global: $\dfrac{104!}{\#\text{double counted}} \quad 2^{52} = \dfrac{104! \text{ all permute}}{\underbrace{2 \cdot 2 \cdot 2 \cdots 2}_{52}}$

$2! = 2$

Constructive

$\underline{\quad} \ \underline{\quad} \ \underline{\quad} \ \underset{3}{\underline{X}} \ \underline{\quad} \ \underset{5}{\underline{X}} \ \underline{\quad} \ \underline{\quad} \ \underline{\quad}$ 104 spots

choose 2 spots for (A ♢ two) $\binom{104}{2} = \dfrac{104 \cdot 103}{2}$

choose 2 spots for (♣ two) $\rightarrow \binom{102}{2} = \dfrac{102 \cdot 101}{2}$

choose 2 spots for (3♡ two) $\rightarrow \binom{100}{2} = \dfrac{100 \cdot 99}{2}$

⋮

choose (last rem 2 spots) for las card ×2 $\rightarrow \binom{2}{2} = \dfrac{2 \cdot 1}{2}$

COUNT 2

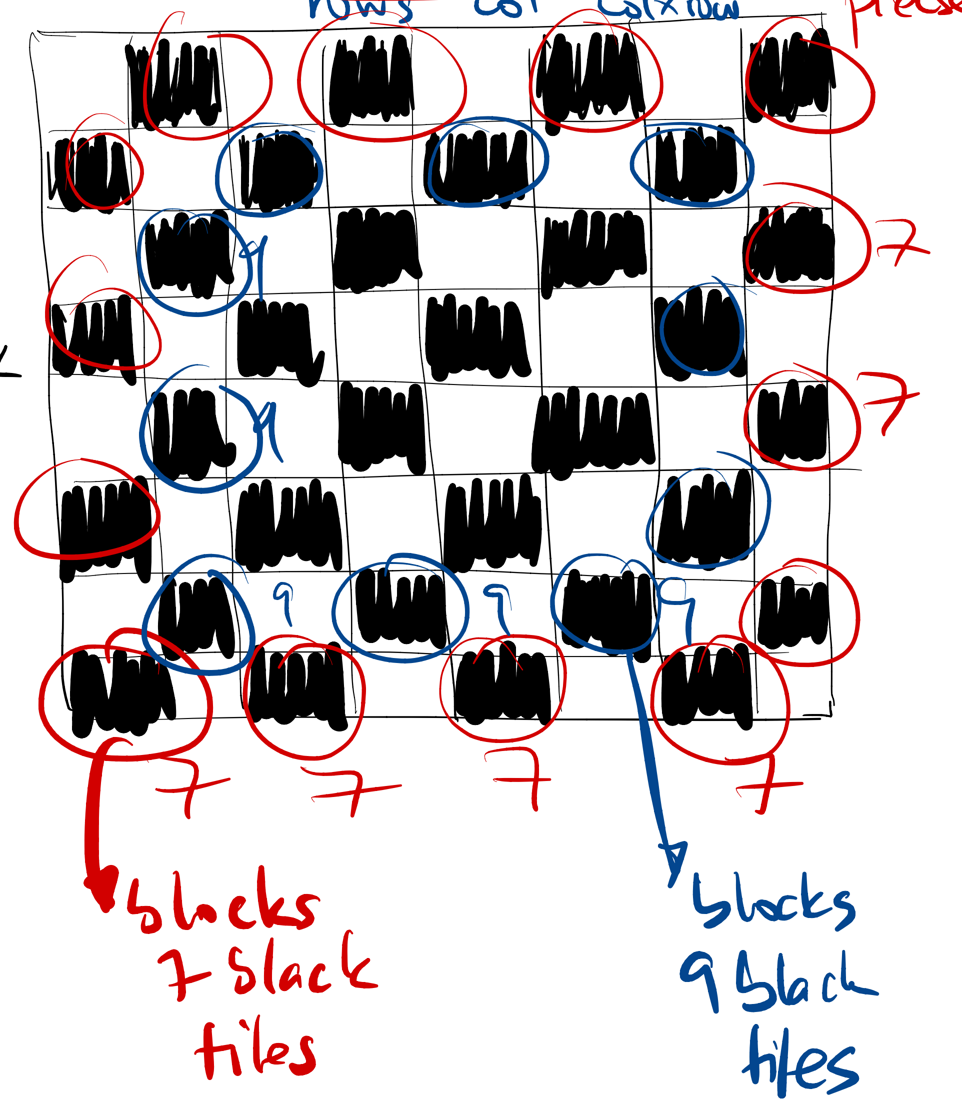
A) How many ways to place a rock, knight and bishop on a chess board ($8\times8$) such that no two of them are on the same row or column?

$\boxed{\underbrace{\binom{8}{3}}_{\text{rows}} \underbrace{\binom{8}{3}}_{\text{col}} \cdot \underbrace{3!}_{\text{col}\times\text{row}}} \times 3!$ please

$8^2 \times 7^2 \times 6^2$ ?

B*) How many ways to place 2 bishops that do not attack each other? [bishop attack on diagonals]

exercise




C) ~~***~~ 3 Bishops

COUNT 3

How many different permutations of "MISSISSIPPI"?

naive: $M_1, S_1, S_2, S_3, S_4, P_1, P_2, I_1, I_2, I_3, I_4$

11 objects: Permute them $\Rightarrow$ 11!

$M_1 S_1 I_1 S_2 I_2 I_3 \ldots$

same $M_1 S_2 I_2 S_1 I_3 I_4 \ldots$ WRONG!

Sol 1: $\dfrac{\text{count all permutations } 11!}{\text{\# permutations make same word}}$

$\boxed{\text{\# permutations } S_i I_i M S_i \ldots}$ $4 \times 4 \times 2$?

all those give same word. $4! \cdot 4! \cdot 2!$?

SIMSISIPISP =
2 5 7 9

I-pos = {2, 5, 7, 9} 4! to place 4-Is on those spots.

## Constructive

- place the S : S S S S 1 choice

  seq 4

- place I-s in between S S S S

  Box 1 Box 5

  4 I-s into 5 boxes

  $\Downarrow$

  seq of of 8 S+I

- place P-s in between

  2 P-s in 9 boxes

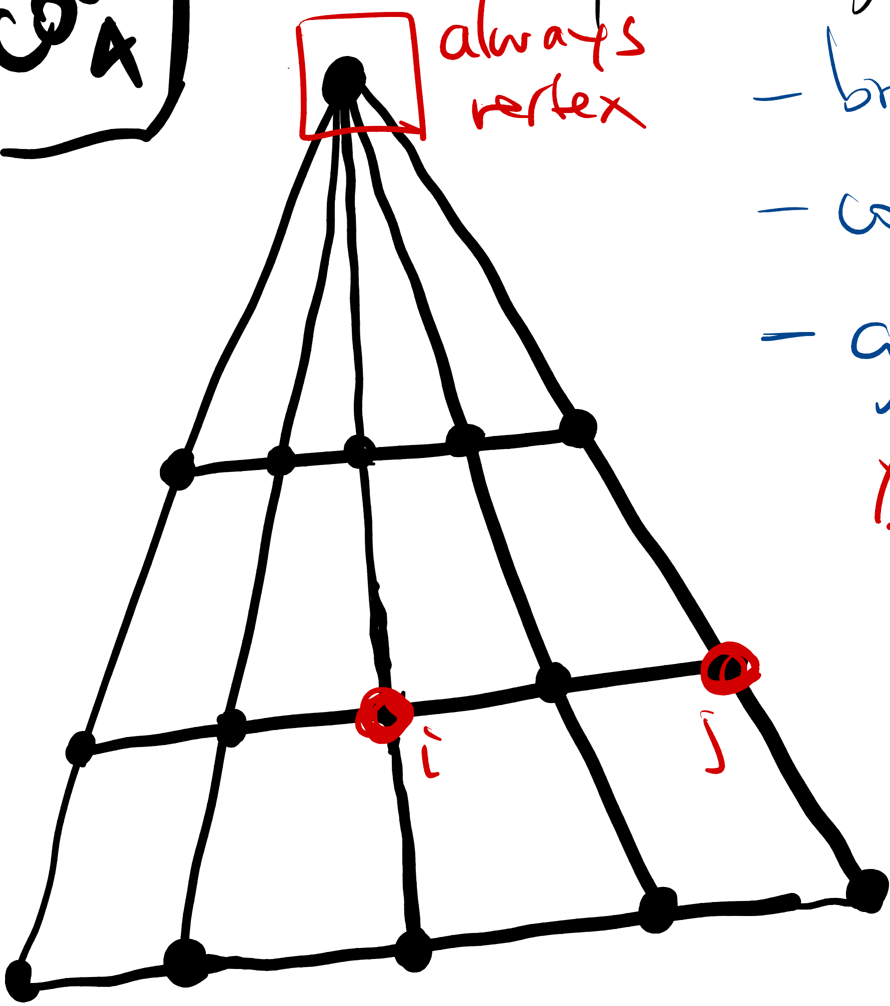
  P+S+I

- place M $\Rightarrow$ 11 choices

- product of choices $\rightarrow$ ? exercise next week

COUNT 4

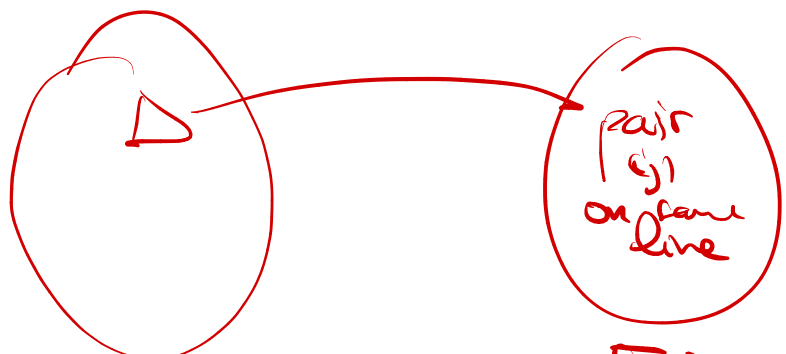
# How many triangles?




Divide & Conquer:

- break pb into choices/subpb
- count/solve those subpb/choices
- glue them together

Δ = choose 2 dots on same horizontal line




$\#$ pairs $(i,j)$ on same line $= \underbrace{3}_{\text{choose a line}} \times \underbrace{\binom{5}{2}}_{\text{choose 2 dots out of 5 (no order, no repet)}} = 3 \cdot \frac{5!}{2!\,3!}$

# $\boxed{\text{Binomial theorem}}$ (coef) $\Rightarrow$ Pascal $\triangle$ $x, y \in \mathbb{R}$

$n=2$ $(x+y)^2 = 1x^2 + 2xy + y^2 = \binom{2}{0}x^2 + \binom{2}{1}xy + \binom{2}{2}y^2$

$(x+y)^3 = 1x^3 + 3x^2y + 3xy^2 + 1y^3 = \binom{3}{0}x^3 + \binom{3}{1}x^2y + \binom{3}{2}xy^2 + \binom{3}{3}y^3$

$(x+y)^4 = 1x^4 + 4x^3y + 6x^2y^2 + 4xy^3 + 1y^4$ $\qquad \binom{4}{2} = \frac{4!}{2!\cdot 2!} = \frac{3\cdot 4}{2}$

$\binom{4}{0}x^4 + \binom{4}{1}x^3y + \binom{4}{2}x^2y^2 + \binom{4}{3}xy^3 + \binom{4}{4}y^4$

$\binom{4}{1}x^{4-1}y^1 \qquad \binom{4}{2}x^{4-2}y^2 \qquad \binom{4}{3}x^{4-3}y^3$

$(x+y)(x+y)(x+y)(x+y) \to$ 16 terms (incl repetion)

4 factors ()

$\to x^4$ — 1 way: x x x x

$\binom{4}{3}$ $\leftarrow$ $\boxed{x^3y^1}$ (3 x, 1 y) xxxy xxyx xyxx yxxx 4

3 x choice

distinct terms $\boxed{x^2y^2}$ xxyy xyxy xyyx yxxy yyxx yxyx 6

$x^1y^3$

$\binom{4}{2}$ choose 2 x

$y^4$

how many ways to choose j out of n

y choice in j paren ()

x choice in n-j paren ()

$$(x+y)^n = \sum_{j=0}^{n} \underbrace{\binom{n}{j} x^{n-j} y^j}_{\text{choose } j \text{ "y"}, \; n-j \text{ "x"}} = \sum_{j=0}^{n} \binom{n}{j} x^j y^{n-j}$$

$2^n$ terms (with repetitions)

$x=1 \quad y=1$

$$2^n = (1+1)^n = \sum_{j=0}^{n} \binom{n}{j} 1^{n-j} \cdot 1^j = \sum_{j=0}^{n} \binom{n}{j} = \binom{n}{0} + \binom{n}{1} + \dots + \binom{n}{n}$$

#subsets $|\,|=0$    #subsets $|\,|=1$    to subset all

$x=+1, \; y=-1$

$$0 = (1-1)^n = \sum_{j=0}^{n} \binom{n}{j} \boxed{1^{n-j}} \underline{(-1)^j} = \binom{n}{0} - \binom{n}{1} + \binom{n}{2} - \binom{n}{3} \dots (-1)^n \binom{n}{n}$$

$n=3 \qquad 1-3+3-1 = 0$

$n=4 \qquad 1-4+6-4+1 = 0$

$n=5 \qquad 1-5+10-10+5-1 = 0$

exercise

$$\binom{n}{k} = \binom{n}{n-k}$$

Pascal △

Binomial coef

| $n-1$ | | | | | | | |
|---|---|---|---|---|---|---|---|
| | 1 | 1 | | | | | $(x+y)^1$ |
| 2 | 1 | 2 | 1 | | | | $(x+y)^2$ |
| 3 | 1 | 3 | 3 | 1 | | | |
| 4 | 1 | 4 | 6 | 4 | 1 | | |
| 5 | 1 | 5 | 10 | 10 | 5 | 1 | |
| 6 | 1 | -6 | +15 | -20 | +15 | -6 | +1 |

exercise

$$\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}$$

$$\binom{5}{3} = \binom{4}{2} + \binom{4}{3}$$

$$\binom{5}{1} = \binom{4}{0} + \binom{4}{1}$$

*

$x = p \quad y = 1-p$ $\quad p =$ prob of success (coin flip)

$1-p =$ prob failure

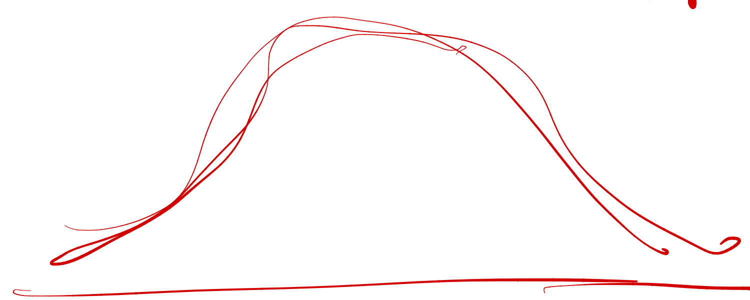
$$1 = (p + (1-p))^n = \sum_{j=0}^{n} \underbrace{\binom{n}{j} p^{n-j} (1-p)^j}_{\text{prob(exactly } j \text{ successes)}}$$ Binomial distribution

$p = \frac{1}{3}$ chance of sum

$n = 100$ coin flips

$$p = \frac{1}{2} : \quad \frac{\sum_{j=0}^{n} \binom{n}{j}}{2^n}$$

dist on piazza ; **Binomial** $n=100, p=\frac{1}{2}$



**gauss approxim** (mean $n/2$, var $n/4$)

good approx for $n \geq 9$