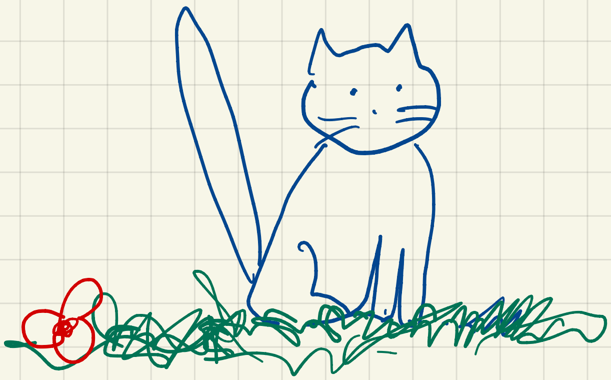
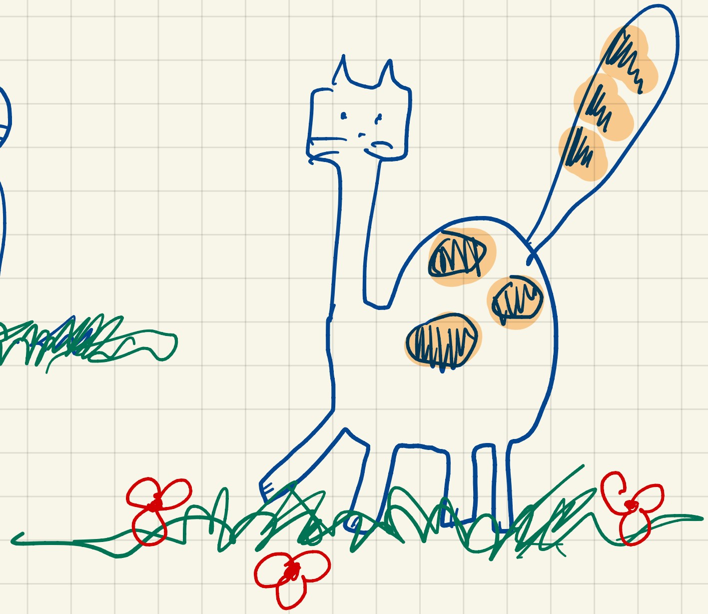
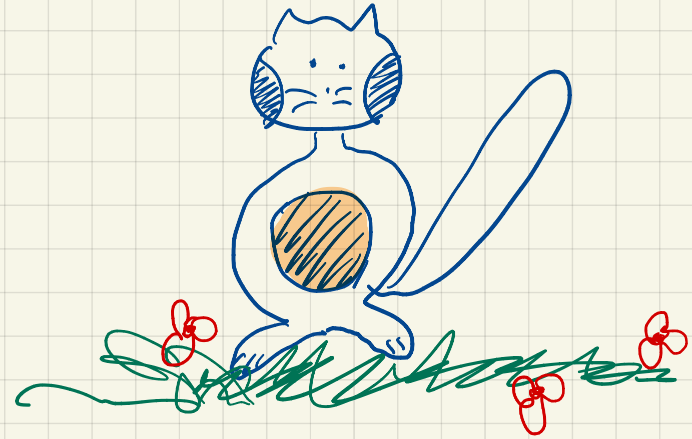
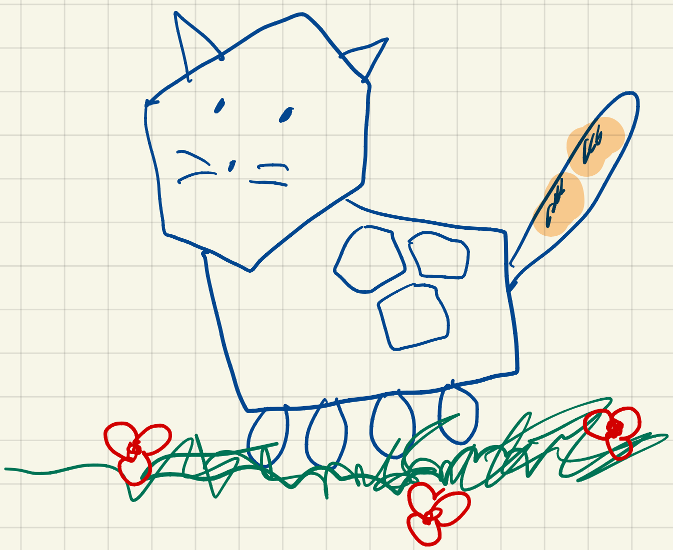
Welcome to Lecture 3

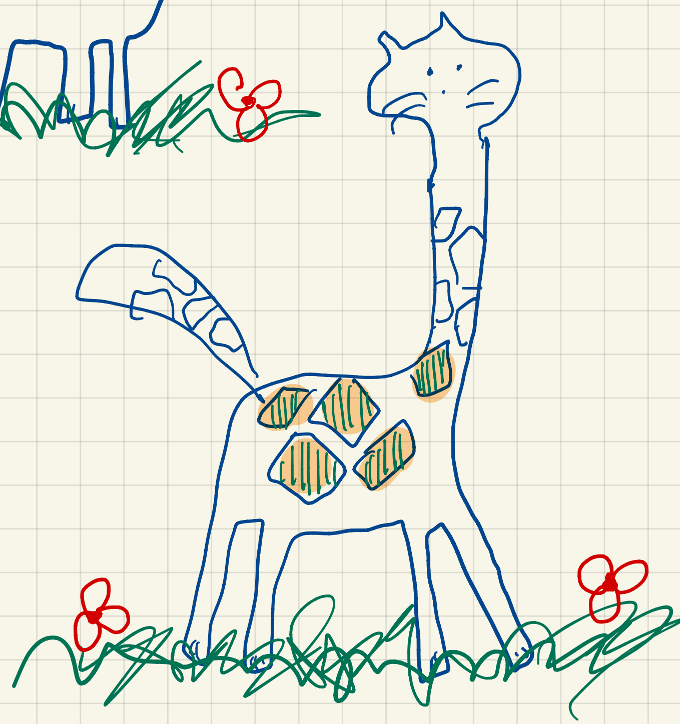
we will start in few minutes



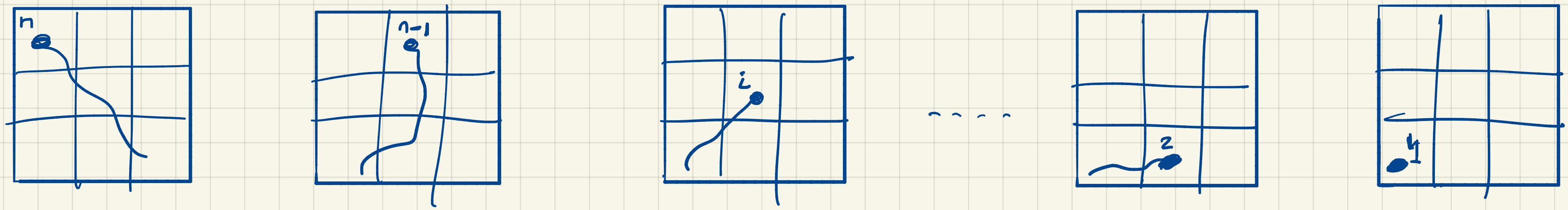








Discrete Math



$$(n-1) \quad + \quad (n-2) \quad + \quad (i-1) \quad + \quad \cdots \quad + \quad 1 \quad + \quad 0$$

Example: $n = 100 \implies 1+2+3+\cdots+99$

We considered disjoint categories of snakes. We can talk about set (Read last Section in chapter 0)

Snakes with head on square n | Snakes with head on square $n-1$ | .... | snakes with head on square 2

S_n | S_{n-1} | | S_2

$|S_n| = n-1$ | $|S_{n-1}| = n-2$ | ..... | $|S_2| = 1$

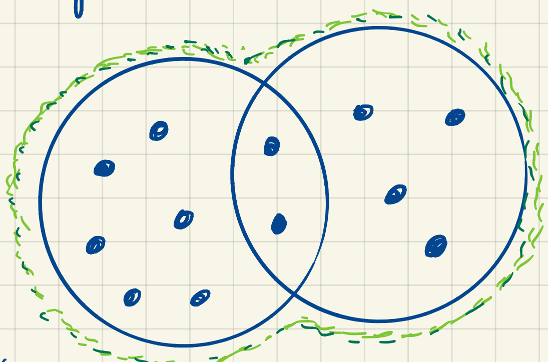
Remark: $|S|$ is the size of set S when S in finite

Addition Rule: Given k sets $S_1, S_2, \dots, S_k$ that are disjoint ($S_i \cap S_j = \phi$), then the total # elements in their union is:

intersection

$$|S_1| + |S_2| + \dots + |S_k| = \sum_{i=1}^{k} |S_i| \qquad (*)$$

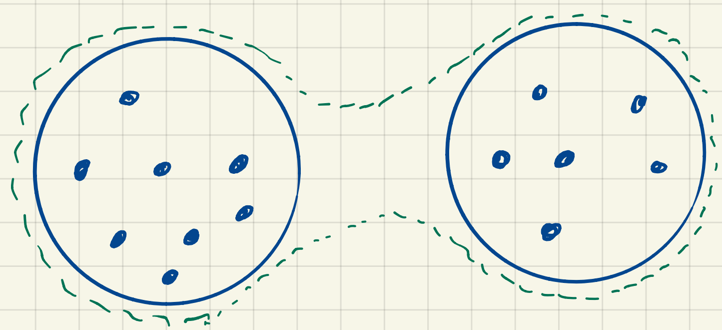
Example:



$|S_1| = 8$ $\qquad |S_2| = 6$

$|S_1 \cup S_2| = 12 \neq 8+6$

union

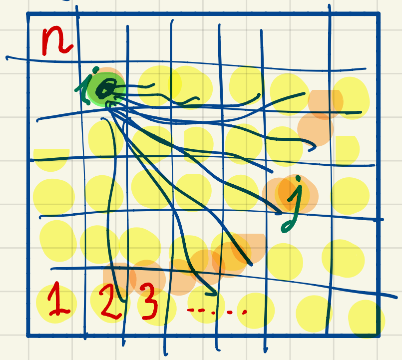


$|S_1| = 8$ $\qquad |S_2| = 6$

$|S_1 \cup S_2| = 14 = 8+6$

(*) why don't we multiply: $|S_1| \times |S_2| \times \dots \times |S_k|$? If we want to choose one from each set then we do.

A program to count all possible placements of one snake



for each head, go through all possible tails

```
s ← 0
for i ← 1 to n        (head)
    for j ← 1 to [?]  (tail)
        do s ← s+1
```

i-1

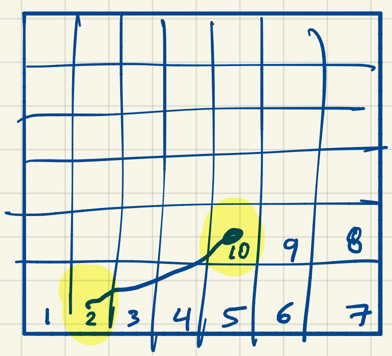
Sum notation:

$$\sum_{i=1}^{n}\left(\sum_{j=1}^{i-1} 1\right) = \sum_{i=1}^{n}(i-1) = \sum_{i=1}^{n} i - \underbrace{\sum_{i=1}^{n} 1} \quad \text{(splitting sum)}$$

$$\frac{n(n+1)}{2} - n = \frac{n(n-1)}{2}$$

or: $\sum_{i=1}^{n}(i-1) = 0+1+2+\cdots+n-1 = n(n-1)/2$

A snake is essentially two squares:



(2, 10)

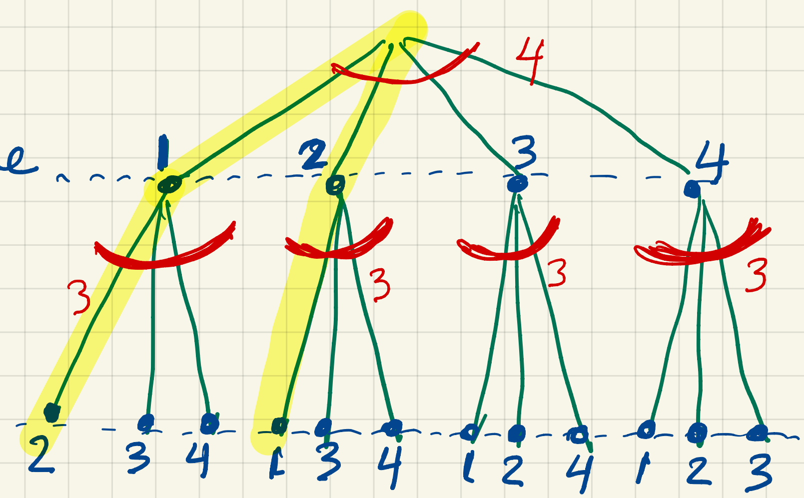
To count something, describe a procedure (procedure must be general enough to produce all) to generate one possible outcome

Example: $n = 4$

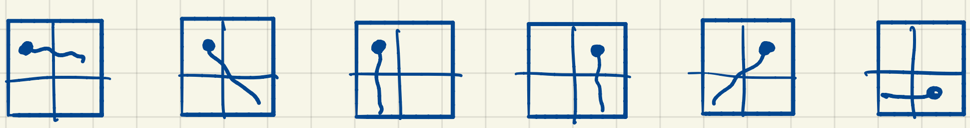
4	3
1	2

Choose 1 square

Choose another square



$4 \times 3 = 12$



$\frac{4 \times 3}{2} = 6$ (over counting by 2)

Product Rule: If a task (procedure) consists of k phases, and phase i can be carried out in α_i ways, INDEPENDENTLY of previous phases, then the entire task can be carried out in

$$\alpha_1 \alpha_2 \alpha_3 \cdots \alpha_k = \prod_{i=1}^{k} \alpha_i \text{ ways.}$$

Snake problem in general

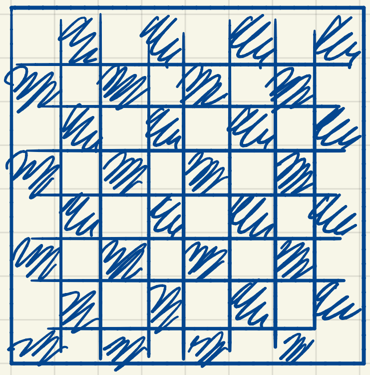
1. choose a square .... n ways
2. choose another square .... $(n-1)$ ways

$$n(n-1)$$

$$\# \text{ snakes} = \frac{n(n-1)}{2} \quad \text{because } (i,j) \equiv (j,i)$$

product rule overcounts by 2 here

Snakes on a chess board (Assume n is even, 64 here)



New rule: head & tail must have same color. (still head > tail)

1. Choose a square -------- n ways

2. Choose another square of the same color ------ $\boxed{\frac{n}{2} - 1}$ ways

$$n\left(\frac{n}{2} - 1\right)$$

Understand independent :

If I choose black in phase 1 I must choose black in phase 2 (dependent ?). But the # ways I can carry out phase 2 is independent of choice in phase 1 !! (so good)

also over counting by 2

ans: $\dfrac{n\left(\frac{n}{2}-1\right)}{2} = \underbrace{\frac{n}{2}\left(\frac{n}{2}-1\right)}$

Another way

(Black snake)

1. Choose a black square .... $\frac{n}{2}$

2. Choose another black square .... $\left(\frac{n}{2}-1\right)$

(White snake)

do the same

(product rule) $\frac{n}{2}\cdot\left(\frac{n}{2}-1\right)$

And since we are overcounting by 2 $\Rightarrow$ $\frac{n}{4}\left(\frac{n}{2}-1\right)$ $\qquad$ $\frac{n}{4}\left(\frac{n}{2}-1\right)$

(Disjoint) $\quad + \quad$ (Addition Rule)

$$\frac{n}{2}\left(\frac{n}{2}-1\right)$$

Place two snakes: (back to regular board, not chess)

Snake 1
1. Choose a square ---- n ways
2. Choose another ----- $n-1$ ways

Snake 2
3. Choose another ----- $n-2$ ways
4. Choose another ------ $n-3$ ways

(product rule) $n(n-1)(n-2)(n-3)/8$

Overcount: Each outcome is counted 8 times

Example: $(1, 2), (3, 4)$

2 2 2

$2 \times 2 \times 2 = 8$

(1,2)	(3,4)
(1,2)	(4,3)
(2,1)	(3,4)
(2,1)	(4,3)