**Math for TOI**

ความน่าจะเป็น

Factorial

0! = 1, 1! = 1, 2! = 2, 3! = 6, n! = n\*(n-1)\*(n-2)\*…\*1

สาเหตุที่ 0! มีค่าเท่ากับ 1 เพราะนักคณิตศาสตร์ต้องการให้ตัว factorial ใช้กับสูตรความน่าจะเป็นได้

กฎการบวก-ลบ

ใช้ในกรณีที่เรารู้ว่ามีวิธีทำงานเสร็จด้วยเงื่อนไขนึง n วิธี และมีวิธีทำงานเสร็จด้วยอีกเงื่อนไข m วิธี จะได้จำนวนวิธีการทำงานเสร็จเท่ากับ n+m \*กฎนี้ใช้ได้เฉพาะงานเสร็จในเงื่อนไขที่ต่างกัน

กฎการคูณ

ใช้ในกรณีที่เราทำงานไปถึงขั้น i เสร็จได้ n วิธี และทำงานขั้น j เสร็จได้ m วิธี แสดงว่ามีวิธีการทำงานเสร็จขั้น i และ j ทั้งสิ้น n\*m วิธี \*กฎข้อนี้ใช้กับการทำงานให้เสร็จเป็นขั้นๆ และเอาแต่ละขั้นมารวมกัน

ความต่างของกฎการคูณและการบวก-ลบ

กฎการคูณจะใช้กับงานที่ยังไม่เสร็จจริงๆ แค่เสร็จเป็นขั้นๆ แต่กฎการบวกนั้นจะเสร็จเรียบร้อย และงานที่เอามารวมกันก็จะต้องเสร็จอยู่ในระดับเดียวกัน แต่เงื่อนไขที่มาต่างกัน

Permutation (nPr) การเรียงสับเปลี่ยน

เป็นปัญหาที่ว่ามีคน n คนมายืนเรียงแถวกัน(ลำดับการมาสำคัญด้วย) ทั้งสิ้น r คน จะเรียงได้ทั้งสิ้น วิธี

Combination (nCr) การเลือก

เป็นปัญหาที่ว่ามีของ n ชิ้นหยิบมาใส่ถุง r ชิ้น(ไม่สนใจลำดับการหยิบใส่ สนว่ามีของชิ้นนั้นอยู่ในถุงไหม) จะทำได้ วิธี

Star & bars

เป็นปัญหาว่ามีดาวเรียงกัน n ดวง และมีไม้อยู่ k อัน จะเอาไม้ไปคั่นระหว่างดาว(จะไม่เอาไม้ไปวางไว้หน้าสุดก่อนดาว หรือหลังสุด) ได้ทั้งสิ้นกี่วิธี โดย star and bars นั้นมี 2 สูตร ดังนี้

กรณีที่จะเอาไม้มาคั่นโดยต้องการให้ระหว่างไม้ 2 อันมีดาวคั่นอยู่อย่างน้อย 1 ดวง จะทำได้ (n-1)C(k-1)

กรณีที่จะเอาไม้มาคั่นเลย จะทำได้ (n+k-1)C(k-1)

ทฤษฎีรังนกพิราบ

บอกว่าถ้ามีรังนก 7 รัง แต่มีนก 8 ตัว แสดงว่ามีอย่างน้อย 1 รังที่มีนกอยู่ 2 ตัวหรือมากกว่า

เกิดเป็นสูตรขึ้นมาคือ ถ้ามีนก n ตัว มีรัง m รัง แสดงว่าจะมีอย่างน้อย n mod m รังที่มีนกมากกว่าหรือเท่ากับ

**Linear Data Structure**

มี 4 ตัวคือ Array, Linked list, Stack, Queue

Array vs Linked List

**การเข้าถึง(access) 1 รอบ**

Array: สามารถเรียกค่าตัวที่ i ได้ใน O(1) เนื่องจาก สามารถเรียกค่า a[i] ออกมาได้ในทันที

Linked list: สามารถเรียกค่าของตัวที่ i ได้ใน O(n) เนื่องจาก ต้องวนหาไล่จากตัวแรกไปตัวต่อไปเรื่อยๆ จนครบ i รอบจึงจะได้

**การแทรกและลบ (Insert & Delete) 1 รอบ**

Array: ทำได้ใน O(n) เพราะต้องไล่วนเลื่อนตัวหลังออก หรือเลื่อนขึ้นมาข้างหน้า

Linked list: ทำได้ใน O(1) เพราะสามารถเปลี่ยนแขนที่ชี้ได้ พูดง่ายๆก็คือสามารถเปลี่ยนการบอกได้ว่าตัวต่อไปเป็นตัวอะไรและตัวก่อนหน้าคือตัวอะไรได้ในทันที

Stack

ข้อมูลที่มีการเข้าถึงจากด้านบนเพียงด้านเดียว Last in First out (LIFO)

Queue

ข้อมูลที่สามารถเข้าถึงได้ 2 ทาง First in First out (FIFO)