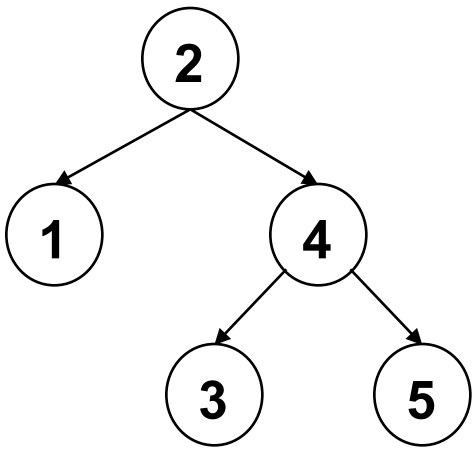
Combinatorics

CS 109
Lecture 2
March 30th, 2016

Binary Search Tree

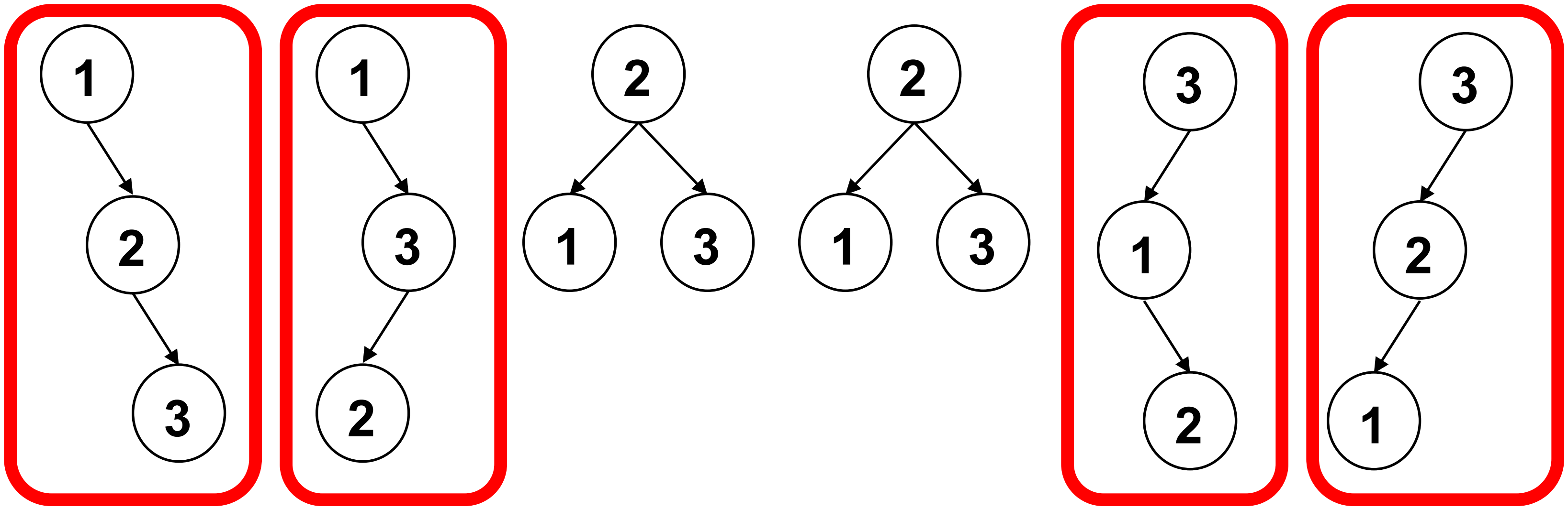
- A **binary search tree** (BST), is a binary tree where for *every* node *n* in the tree:
  - *n*'s value is **greater** than all the values in its **left** subtree.
  - *n*'s value is **less** than all the values in its **right** subtree.
  - both *n*'s left and right subtrees are binary search trees.



Binary Search Tree

- **Problem**: How many possible BSTs containing values 1, 2, and 3 have degenerate structure (i.e., each node in the BST has at most one child)?
- **Solution**: 3! ways to order 1, 2, and 3 for insertion

1, 2, 3   1, 3, 2   2, 1, 3   2, 3, 1   3, 1, 2   3, 2, 1



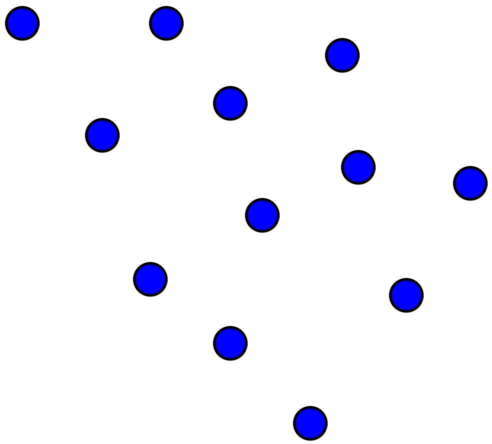
There are 4 degenerate BSTs possible.

Recursive definition of $\binom{n}{k}$

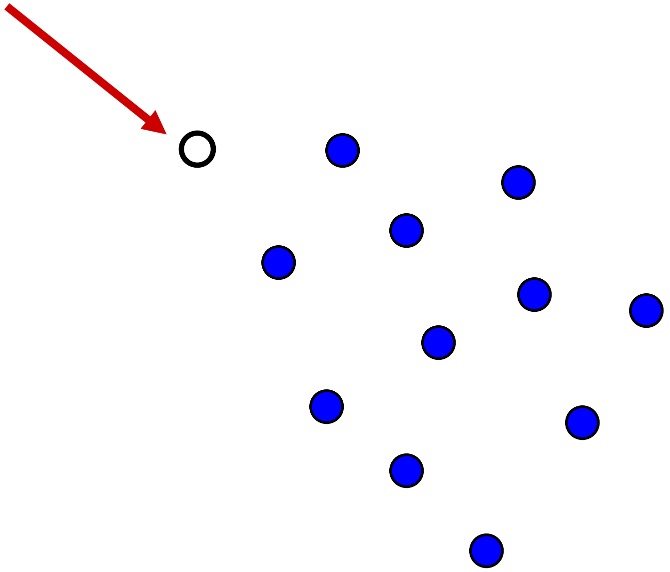
Let's write a function C(n, k)

The number of ways to select **k** objects from a set of **n** objects.

C(n,k)

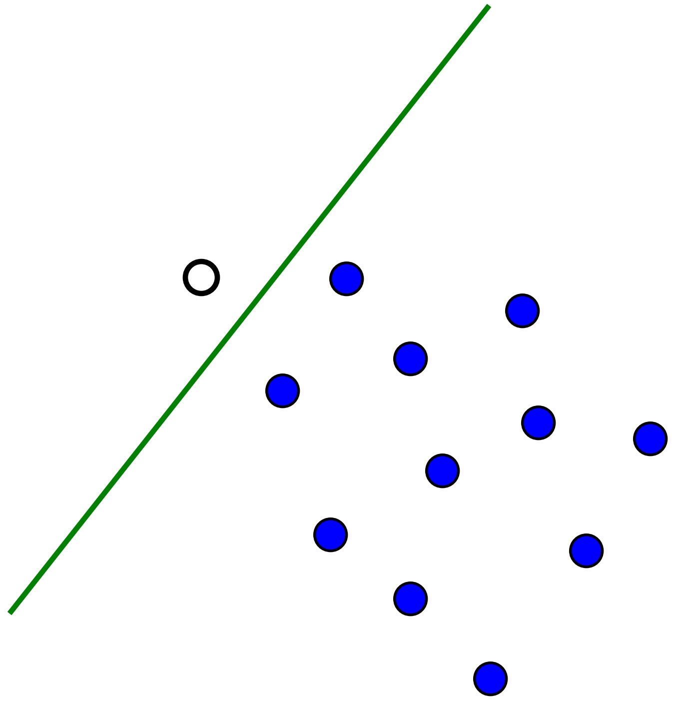


C(n,k)



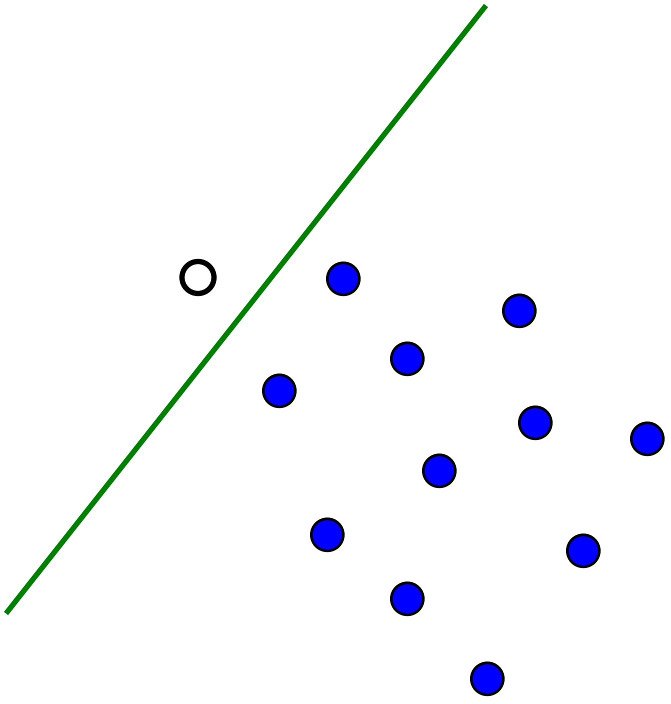
Select any one of the n points in the group

C(n,k)



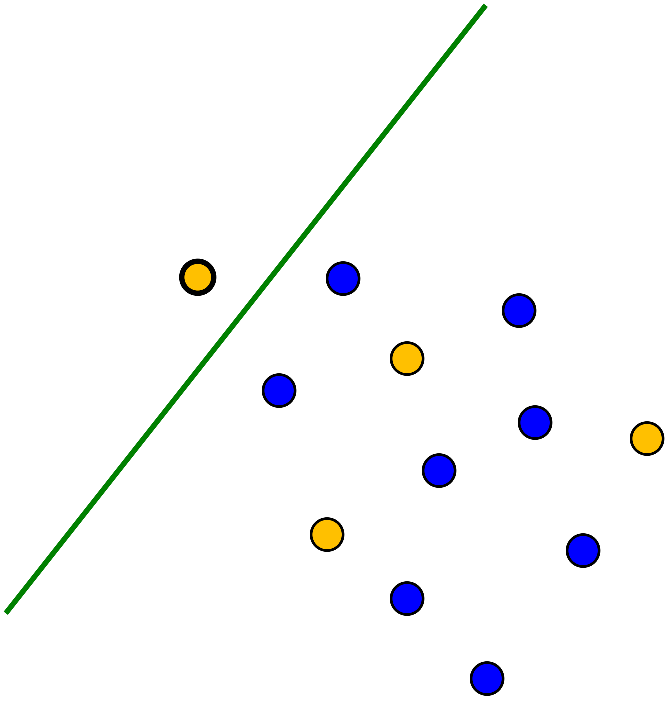
Separate this point from the rest

C(n,4)



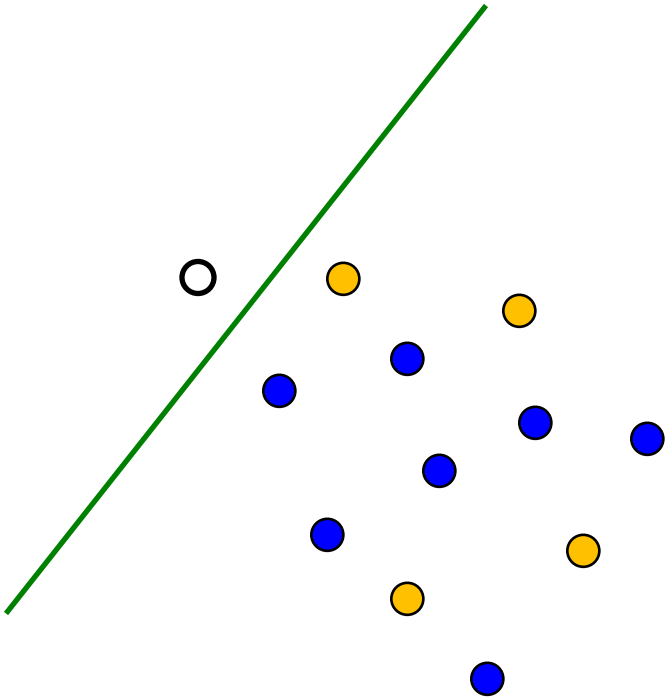
Let’s consider specific problem C(n, 4)

C(n,4)



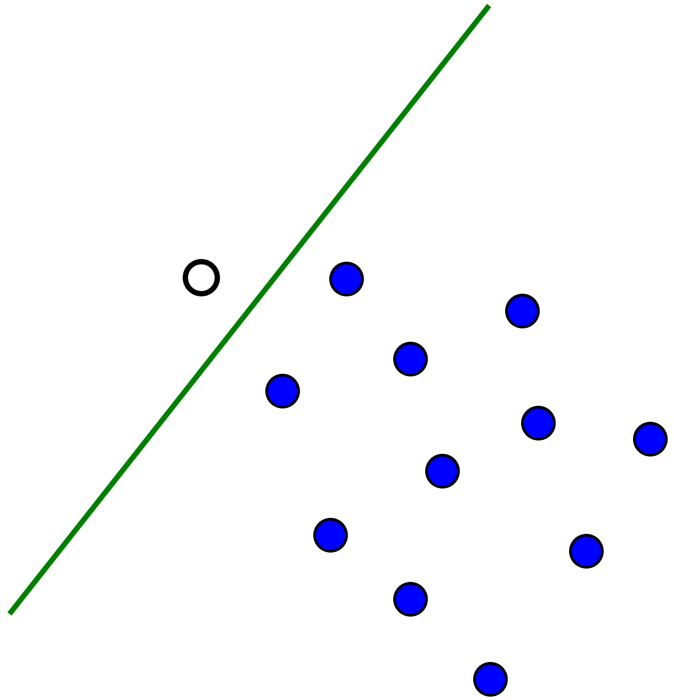
This point can be **included** in the 4 points we choose

C(n,4)



Or, it can be **excluded** from the 4 points we choose

C(n,k)



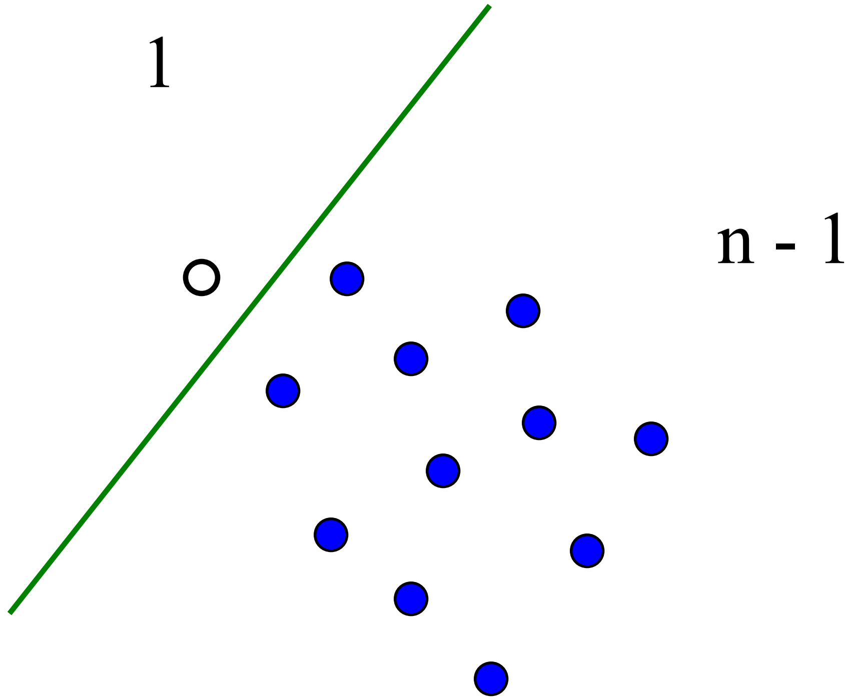
Total number of solutions is

number of solutions including ○

+

number of solutions not including ○

C(n,k)



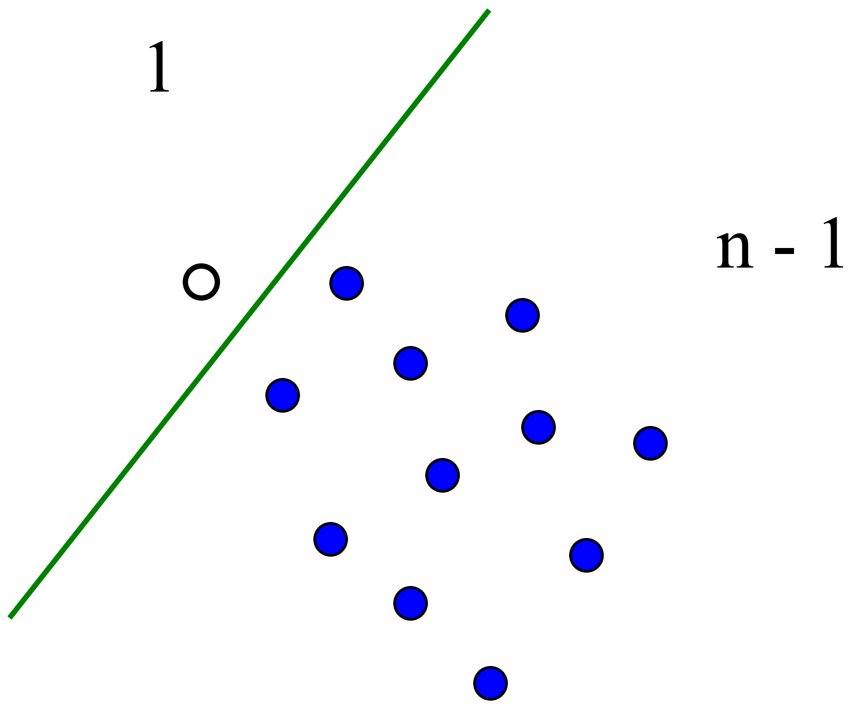
Total number of solutions is

number of solutions including ○

+

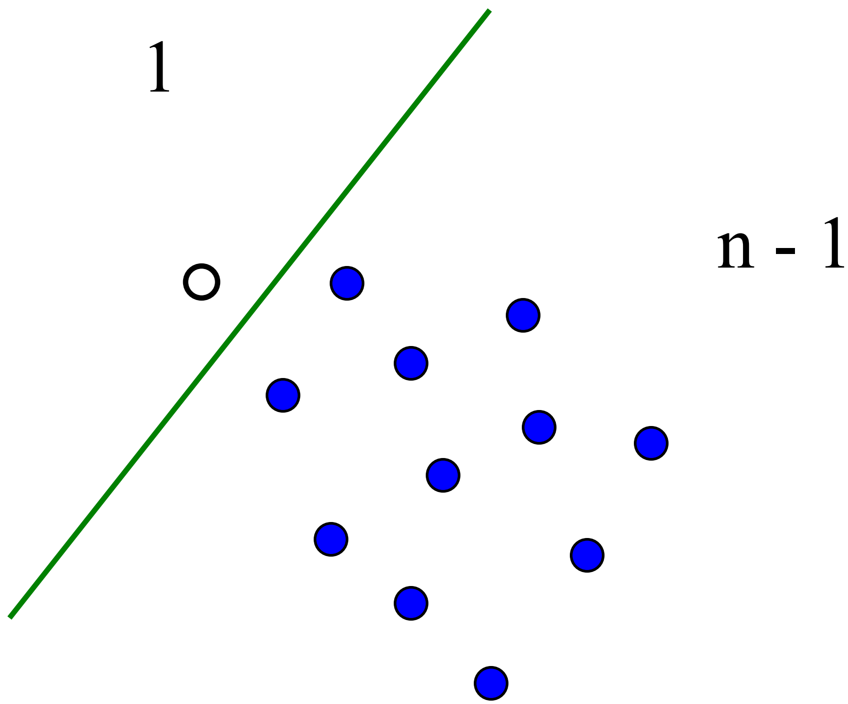
number of solutions not including ○

C(n,k)



number of solutions including ○ `C(n-1, k-1)`

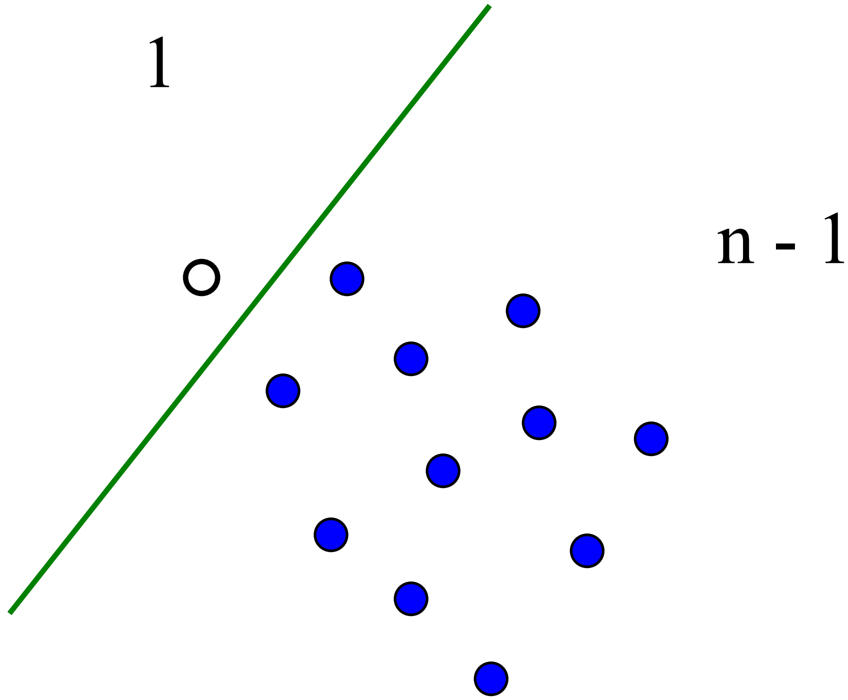
C(n,k)



number of solutions including ○ `C(n-1, k-1)`

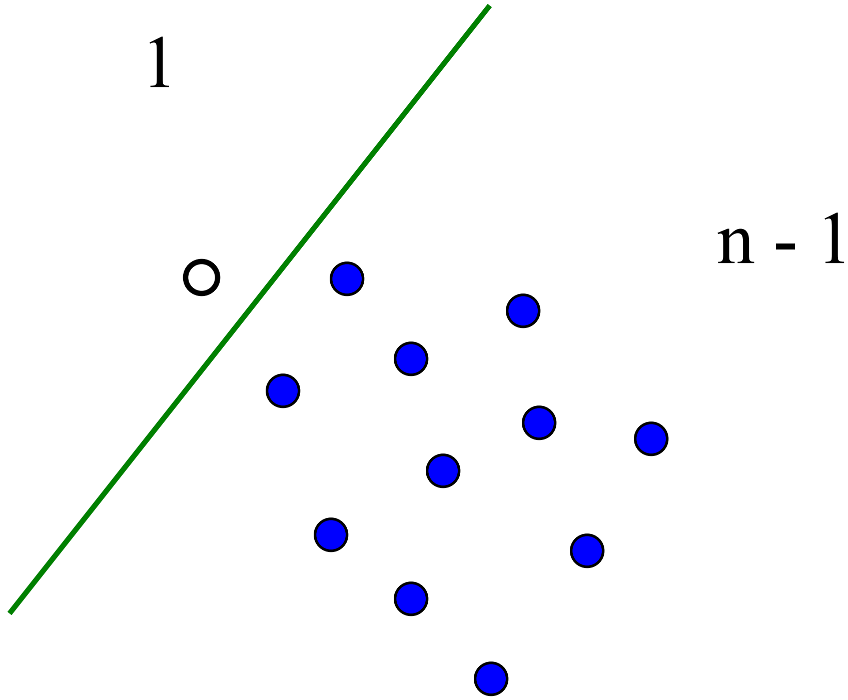
number of solutions not including ○ `C(n-1, k)`

C(n,k)



<u>Total number of solutions is</u> **`C(n-1, k-1)  +  C(n-1, k)`**

C(n,k)




```
int C(int n, int k)
{
    if (k == 0 || n == k) return (1);
    return (C(n-1, k-1) + C(n-1, k));
}
```