Combinatorics and Probability from Foundations of Computer Science by Aho and Ullman, Chap 4 and other notes. The bucket distribution notes are largely from https://math.berkeley.edu/~evans/Combinatorics which are 2013 lecture notes by Lior Pachter and Lawrence C. Evans, UC, Berkeley, “Methods Of Mathematics: Calculus, Statistics And Combinatorics” Also see https://dlmf.nist.gov/26
Not used directly here, but frequently referenced in the material that is: “The On-Line Encyclopedia of Integer Sequences”, http://oeis.org/wiki/Main_Page

A **multiset** (or **bag**, or **mset**) is a modification of the concept of a set that, unlike a set, allows for multiple instances for each of its elements.

A **partition** of an integer n is a multiset (or bag) of positive integers whose elements sum to n . This is an additive representation of n . A part in a partition is sometimes also called a **summand.** **The set of partitions of n is denoted by P (n)** . The partition function **p (n) gives the number of partitions of n**, that is p (n) is the cardinality of P (n) .

The set of partitions of 0 is an empty bag: P(0) = {∅} = {{ }}, p(0) = 1.
The set of partitions of a negative integer is the empty set, since neg. Integers are not the sum of positive. for n < 0: P(n) = ∅ = { }, p(n) = 0.

https://en.wikipedia.org/wiki/Partition_(number_theory)
No closed-form expression for the partition function is known, but it has both asymptotic expansions that accurately approximate it and recurrence relations by which it can be calculated exactly. It grows as an exponential function of the square root of its argument.[3] The multiplicative inverse of its generating function is the Euler function; by Euler's pentagonal number theorem this function is an alternating sum of pentagonal number powers of its argument.

$p_k(n)$ denotes the number of permutations of n into at most **(arbitrary)** k parts:

$$\mathbf{p_k(n) = p_{k-1}(n-1) + p_k(n-k)}$$

$p_k(n) = 0$ if $k > n$

$p_1(n) = p_n(n) = 1$; $p_k(n=0) = 1$; $p_0(n=0)=1$; $p_1(n)=1$; $p_0(n>=1)=0$

e.g. n=4, k=3:

$$\begin{aligned} p_3(4) &= p_2(3) + p_3(1) \\ &= (p_1(2) + p_2(1)) + (p_2(0) + 0) \\ &= p_0(1) + p_1(1) + p_1(0) + p_2(0) \\ &= 1 + 1 + 1 + 1 \end{aligned}$$

arbitrary k

Table 26.9.1: Partitions $p_k(n)$.

n \ k	0	1	2	3	4	5	6	7	8	9	10
0	1	1	1	1	1	1	1	1	1	1	1
1	0	1	1	1	1	1	1	1	1	1	1
2	0	1	2	2	2	2	2	2	2	2	2
3	0	1	2	3	3	3	3	3	3	3	3
4	0	1	3	4	5	5	5	5	5	5	5
5	0	1	3	5	6	7	7	7	7	7	7
6	0	1	4	7	9	10	11	11	11	11	11
7	0	1	4	8	11	13	14	15	15	15	15
8	0	1	5	10	15	18	20	21	22	22	22
9	0	1	5	12	18	23	26	28	29	30	30
10	0	1	6	14	23	30	35	38	40	41	42

https://dlmf.nist.gov/26.9#T1

A **lattice path** in the plane is a curve made up of line segments that either go from a point (i, j) to the point (i + 1, j) or from a point (i, j) to the point (i, j + 1) where i and j are integers. (Thus lattice paths always move either up or to the right.)
https://www.math.toronto.edu/balazse/2019_Summer_MAT344/Lec_4.pdf

A **permutation** is an ordered arrangement of n distinct objects.
A **combination** is an unordered selection of r objects from a set of n objects.

— The number of ways to arrange n distinct items is ***n!***
— The number of assignments of n values to k objects is n^k
(e.g. 5 slots of 0 or 1 is 2^5 = 32 possible assignments)
This is ordered, w/ replacement.
— For ordered, w/o replacement:
The number of ways to select a sequence (not a set) of k items out of n distinct items for a fixed length of k, i.e. k-permutations of n, i.e. such that [a,b] and [b,a] are counted instead of only {a,b}:
n! / (n − k)!.

— The number of ways to arrange n indistinct items is estimated using product rule:
permutation of distinct objects =
permutations considering some are indistinct **X** permutations of only indistinct objects
==> permutations considering some are indistinct =
permutation of distinct / permutations of only indistinct
= *n! / k!*
Permutation of Indistinct Objects when there are n objects and n_1 are the same (indistinguishable), n_2 are the same, ... and n_r are the same, then there are ***n! / (n_1! * n_2! * n_3! ... n_r!)*** **distinct permutations** of the objects.
This is also called a **partition rule**. **multisite permutations** can also be estimated this way (i.e. misspelt missippi = 11!/((3!)*(2!)*(2!))).

— The number of ways to select k sub**sets** out of n distinct objects, is $\binom{n}{k}$ = ***n! / (k!(n − k)!)***
also stated as: select k unordered objects from a set of n objects. This is **k-combinations, C(n, k)**.

-- The number of multisets of cardinality k, with elements taken from a finite set of cardinality n, is called the multiset coefficient or multiset number. This could be pronounced "n multichoose k" to resemble "n choose k", or *k*-**combination with repetitions**, or *k*-**multicombination.** The value of multiset coefficients is

$$\left(\!\!\binom{n}{k}\!\!\right) = \binom{n+k-1}{k}$$

The distribution of objects into containers:

Adapted from https://math.berkeley.edu/~evans/Combinatorics

Given a collection of containers and a number of objects to put in them:

there are 3 different rules for mapping/assigning the objects to containers:
- injective: at most 1 object in each container
- subjective: at least 1 object in each container
- no restrictions on the number of objects per container

there are 2 possibilities regarding the collection of objects
- **distinct** or **indistinct** from each other

there are 2 possibilities regarding the containers
- **distinct** or **indistinct** from each other

Therefore, 3*2*2=*12 different types of counting problems*.

table summarizing the methods, from https://dlmf.nist.gov/26.17

n objects k containers arbitrary k injective k <= 1 subjective k >= 1

Table 26.17.1: The twelvefold way.

elements of N	elements of K	f unrestricted	f one-to-one	f onto
labeled	labeled	k^n	$(k-n+1)_n$	$k!\,S(n,k)$
unlabeled	labeled	$\binom{k+n-1}{n}$	$\binom{k}{n}$	$\binom{n-1}{n-k}$
labeled	unlabeled	$S(n,1)+S(n,2)+\cdots+S(n,k)$	$\begin{cases}1 & n\le k\\ 0 & n>k\end{cases}$	$S(n,k)$
unlabeled	unlabeled	$p_k(n)$	$\begin{cases}1 & n\le k\\ 0 & n>k\end{cases}$	$p_k(n)-p_{k-1}(n)$

^Where $(k)_n$ is Pochhammer's symbol = k*(k+1)*...(k+n-1).

^Where S(n, k) is the Stirling numbers of the second kind S(n, k) count the number of ways to partition a set of n objects into k nonempty (and indistinct) subsets. The Stirling numbers satisfy the recurrence S(n+1, k) = k*S(n, k) + S(n, k−1), (1)
with S(0, 0) = 1 and S(n, 0) = S(0, k) = 0.

$$S(n,k) = \frac{1}{k!}\sum_{j=0}^{k}(-1)^{k-j}\binom{k}{j}j^n.$$

https://dlmf.nist.gov/26.8

^Where $p_k(n)$ is the partition function. $p_k(n)$ counts the number of partitions of n into k parts (the number of distinct ways to write n as the sum of k positive integers). Note that

$p_k(n) = p_{k-1}(n-1) + p_k(n-k)$

where $p_1(n) = p_n(n) = 1$ and $p_k(n) = 0$ if $k > n$.

$$p_k(n) = \frac{1}{n}\sum_{t=1}^{n} p_k(n-t)\sum_{\substack{j|t \\ j\le k}} j,$$

Bucketing **n** **<u>distinct</u> objects** into <u>distinct</u> **k containers**:

— The number of ways = ***k^n***

Bucketing **n** **<u>distinct</u> objects** into <u>distinct</u> **k containers, with no more than 1 object per container (<u>injective</u>)**:

— The number of ways = ***$k\cdot(k - 1)\cdots(k - n + 1)$***

(NLK: does this assume k>n? Full formula = k!/n! before simplification)

Bucketing **n** **<u>indistinct</u> objects** into <u>distinct</u> **k containers, with no more than 1 object per container (<u>injective</u>)**: uses the product rule.

— The number of ways = ***$k\cdot(k - 1)\cdots(k - n + 1) / n!$***

Which can be expressed with combinatorial symbol ***$C(k, n)$***.

(NLK: similar to the above, but "indistinct objects" reduces the permutation to unique set, i.e. (a,b) and (b,a) are only {a,b} so the number of ways is smaller. Smaller by the amount of n sub**sets** out of k distinct containers)

Bucketing **n** **<u>indistinct</u> objects** into <u>distinct</u> **k containers, with no restrictions on distribution**:

— The number of ways = ***$C(k+n-1, n)$***

Bucketing **n** **<u>indistinct</u> objects** into <u>distinct</u> **k containers, with at least one object in each container (subjejunctive)**

— The number of ways = ***$C(n-1, n-k)$***

Bucketing **n** **<u>distinct</u> objects** into <u>indistinct</u> **k containers, with at least one object in each container (subjunctive)**

— The number of ways = ***$S(n, k)$***

Bucketing **n** **<u>distinct</u> objects** into <u>indistinct</u> **k containers, with no more than 1 object per container**

— The number of ways = ***1 if $n \le k$ and 0 if $n > k$***

Bucketing **n** **<u>distinct</u> objects** into <u>indistinct</u> **k containers, with no restrictions on distribution**

— The number of ways = ***$\sum_{i=1,k} S(n,i)$***

n **<u>indistinct</u>** objects in the k **<u>indistinct</u>** containers?

As the objects and containers are indistinct, this is equivalence partitioning of a natural number n into a sum of k natural numbers.

Bucketing **n** **<u>indistinct</u> objects** into <u>indistinct</u> **k containers, with at least one object in each container (subjunctive).**

— The number of ways = ***$p_k(n)$***

Also phrased as Partition n into a sum of k positive integers.

Bucketing **n** **<u>indistinct</u> objects** into <u>indistinct</u> **k containers, with no more than 1 object per container**

— The number of ways = ***1 if $n \le k$ and 0 if $n > k$***

Bucketing **n <u>indistinct</u> objects** into <u>indistinct</u> **k containers, with no restrictions on distribution**

— The number of ways = $\sum_{i=1,k} p_k(n)$

Also phrased as Partition n into a sum of at most k positive integers.

More partitions:

— The number of ways to place n indistinct objects into k containers where the n items are of m classes

(n + k - 1)! / ((k - 1)! * products of each size class)

— The length of a <u>lattice path</u> from (0, 0) to (m, n) = m + n

— The number of possible lattice paths from (0, 0) to (m, n) of the shortest length, m + n,

Is to choose m from the m+n = C(m+n, m)

— The number of possible lattice paths from (i, j) to (m, n) , assuming i, j, m, n are integers:

If i > m or j > n then we can not go from (i, j) to (m, n), since we can only travel up and right.

Otherwise we have to make m − i right and n − j up steps = C(m-i+n-j, m-i)

Combinatorics and Probability (cont.)

— **<u>Bayes Rule:</u>**

P(B|A) = P(B)*P(A|B) / P(A)

prior is P(B)

posterior is P(B|A)

likelihood is P(A|B)

normalizing constant is P(A)

— **<u>MAP (maximum-a-Posteriori estimation):</u>**

Choose θ (== parameters) that maximizes the posterior probability of θ

MLE estimation of a parameter leads to unregularized solutions.

MAP estimation of a parameter leads to regularized solutions.

The prior distribution acts as a regularizer in MAP estimation

Note: For MAP, different prior distributions lead to different regularizers

Gaussian prior on w regularizes the ℓ2 norm of w

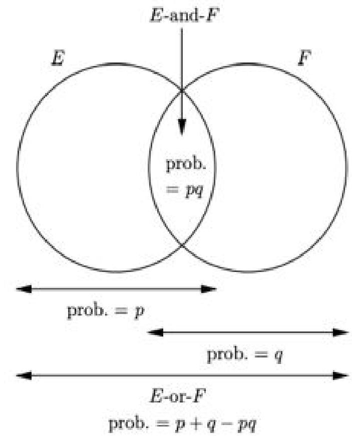
Laplace prior exp (−C||w||1) on w regularizes the ℓ1 norm of w

Combinatorics and Probability (cont.)

Summary of Rules Involving Several Events

The following summarizes rules of this section and rules about independent events from the last section. Suppose E and F are events with probabilities p and q, respectively. Then

- The probability of event E-or-F (i.e., at least one of E and F) is at least $\max(p, q)$ and at most $p + q$ (or 1 if $p + q > 1$).
- The probability of event E-and-F (i.e., both E and F) is at most $\min(p, q)$ and at least $p + q - 1$ (or 0 if $p + q < 1$).
- If E and F are independent events, then the probability of E-and-F is pq.
- If E and F are independent events, then the probability of E-or-F is $p+q-pq$.



<u>The geometric and binomial distributions</u>:

Bernoulli Trials have only success or failure as the possible outcome.
p is the probability of success.
q is the probability of failure. q = 1 - p.
The number of trials before a success is k.

The geometric probability distribution is

$P[X=k] = q^{k-1} * p = (1-p)^{k-1} * p$

It has E[X] = 1/p

And Var[X = q / p^2

The binomial probability distribution estimates the number of successes for n Bernoulli Trials.

$P[X=k] = C(n, k) * p^k * (1-p)^{n-k}$

It has E[X] = n *p

And Var[X = n * p * q