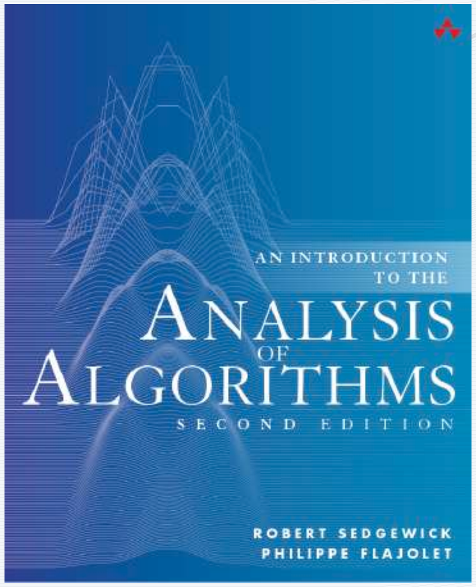
ANALYTIC COMBINATORICS

PART ONE



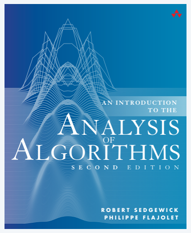
http://aofa.cs.princeton.edu

7. Permutations

Orientation

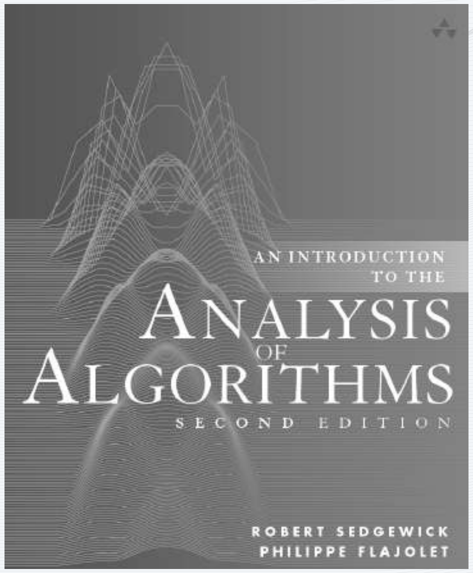
Second half of class

- Surveys fundamental combinatorial classes.
- Considers techniques from analytic combinatorics to study them .
- Includes applications to the analysis of algorithms.



chapter	*combinatorial classes*	*type of class*	*type of GF*
6	Trees	unlabeled	OGFs
7	Permutations	labeled	EGFs
8	Strings and Tries	unlabeled	OGFs
9	Words and Mappings	labeled	EGFs

Note: Many more examples in book than in lectures.



http://aofa.cs.princeton.edu

7. Permutations

- **Basics**
- Sets of cycles
- Left-right-minima
- Other parameters
- BGFs and distributions

Basics

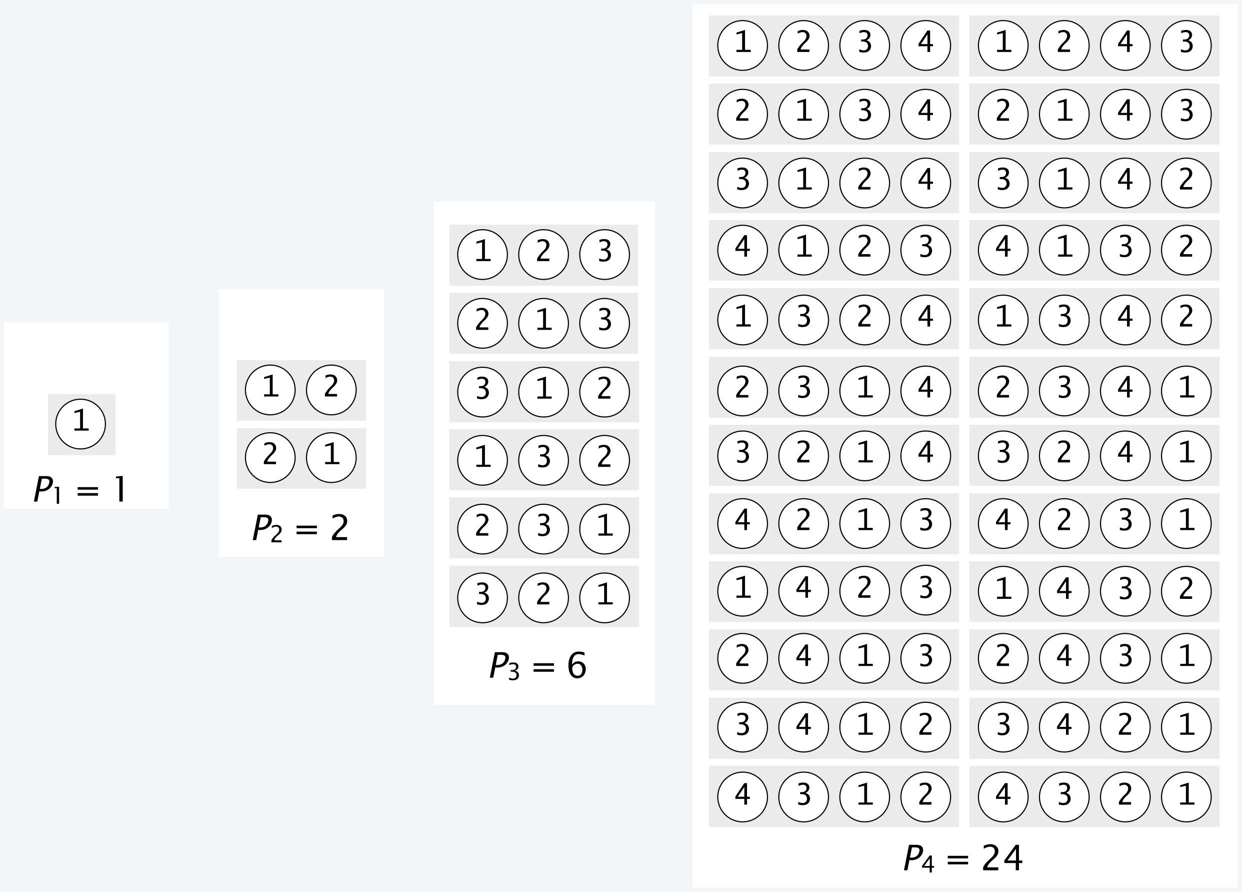
Definition. A permutation is an ordering or the numbers 1 through *N*.

Ex. A group of *N* students who live in single rooms go to a party that leads to a state of inebriation. When returning, they each end up in a random room.

student	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
room	9	12	11	10	5	15	1	3	7	6	13	8	2	16	4	14

Review: permutations

Def. A *permutation* is a sequence of labelled atoms.



$P_1 = 1$ $P_2 = 2$ $P_3 = 6$ $P_4 = 24$

counting sequence	*EGF*
$P_N = N!$	$\dfrac{1}{1-z}$

$$\sum_{N\geq 0} \frac{N! z^N}{N!} = \sum_{N\geq 0} z^N = \frac{1}{1-z}$$

Inverse

Alternate def. A *permutation* is a mapping of the numbers 1 through *N* to itself.

student	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
room	9	12	11	10	5	15	1	3	7	6	13	8	2	16	4	14

Def. The *inverse* of a permutation is the inverse of that mapping.

student	7	13	8	15	5	10	9	12	1	4	3	2	11	16	6	14
room	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

room	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
student	7	13	8	15	5	10	9	12	1	4	3	2	11	16	6	14

Computing the inverse of a permutation

```
public static int[] inverse(int[] a)
{
   int N = a.length;
   int[] b = new int[N];
   for (int i = 0; i < N; i++)
      b[a[i]-1] = i+1;
   return b;
}
```

Java arrays are 0-based

permutation

1	2	3	4	5	6	7	8	9
8	1	3	7	6	2	9	4	5

inverse

							1	
2							1	
2		3					1	
2		3				4	1	
2		3			5	4	1	
2	6	3			5	4	1	
2	6	3			5	4	1	7
2	6	3	8		5	4	1	7
2	6	3	8	9	5	4	1	7

Application: Substitution cipher

Algorithm (traditional)

- Generate random permutation of A-Z (stay tuned).
- Apply as a mapping to encrypt.
- Use inverse to decrypt.

Encryption

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	-
random permutation	W	V	L	Q	I	X	J	A	B	G	-	U	N	F	K	R	Y	C	D	P	Z	E	O	M	H	T	S

plaintext	A	T	T	A	C	K	-	A	T	-	D	A	W	N
ciphertext	W	P	P	W	L	-	S	W	P	S	Q	W	O	F

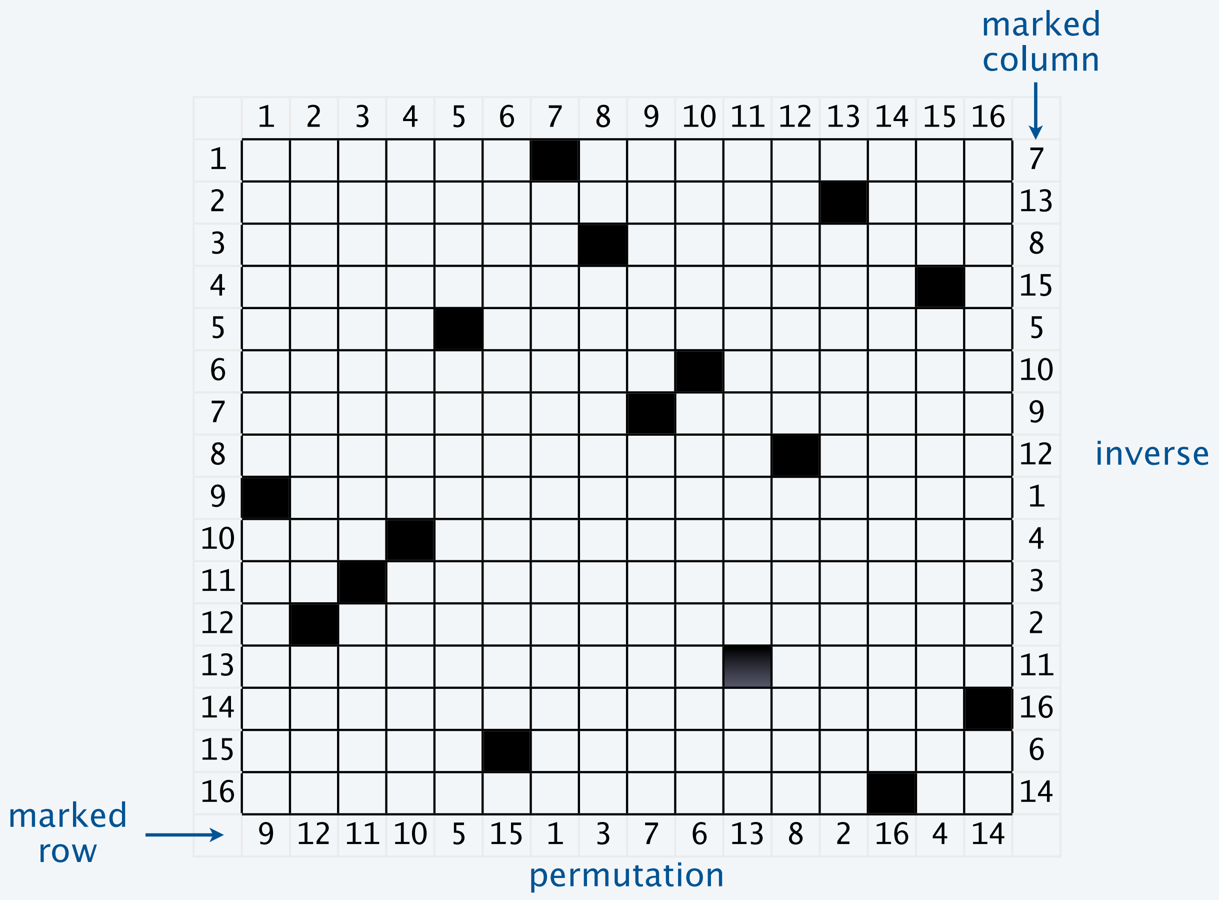
Decryption

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	-
inverse	H	I	R	S	V	N	J	Y	E	G	O	C	X	M	W	T	D	P	-	Z	L	B	A	F	Q	U	K

ciphertext	W	P	P	W	L	-	S	W	P	S	Q	W	O	F
plaintext	A	T	T	A	C	K	-	A	T	-	D	A	W	N

Caveat. Not useful in modern applications because of susceptibility to character frequency analysis.

Lattice representation of a permutation

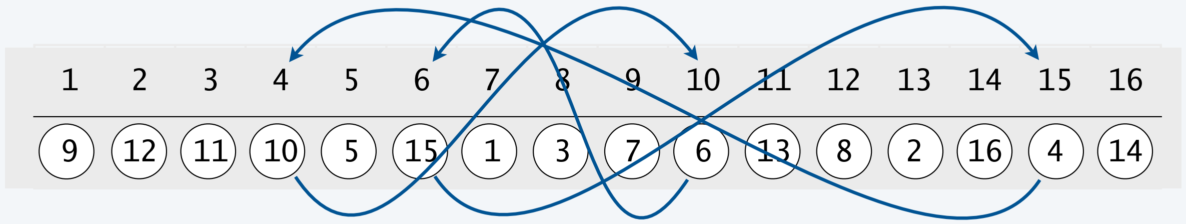


Implication. Representation of inverse is *transpose* of representation of permutation.

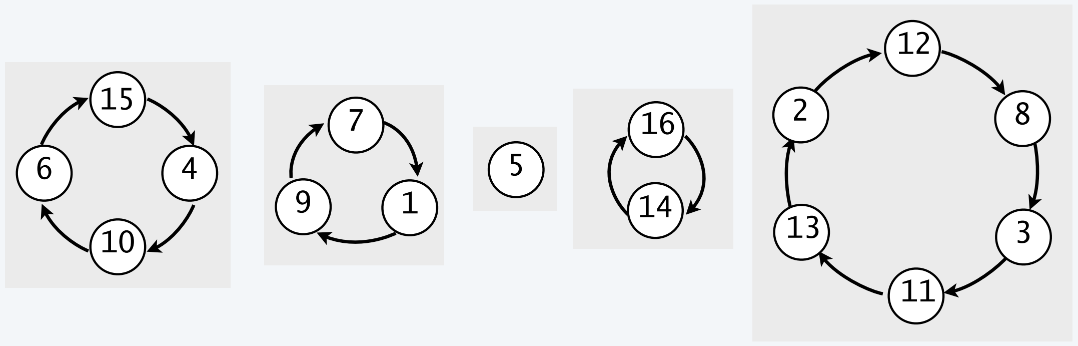
Review: A combinatorial bijection

Alternate def. A permutation is a set of cycles.

Standard representation



Set of cycles representation



Review: The symbolic method for labelled classes (transfer theorem)

Theorem. Let A and B be combinatorial classes of labelled objects with EGFs $A(z)$ and $B(z)$. Then

construction	*notation*	*semantics*	*EGF*
disjoint union	$A + B$	disjoint copies of objects from A and B	$A(z) + B(z)$
labelled product	$A \star B$	ordered pairs of copies of objects, one from A and one from B	$A(z)B(z)$
sequence	$SEQ_k(A)$	k- sequences of objects from A	$A(z)^k$
	$SEQ(A)$	sequences of objects from A	$\frac{1}{1-A(z)}$
set	$SET_k(A)$	k-sets of objects from A	$A(z)^k/k!$
	$SET(A)$	sets of objects from A	$e^{A(z)}$
cycle	$CYC_k(A)$	k-cycles of objects from A	$A(z)^k/k$
	$CYC(A)$	cycles of objects from A	$\ln \frac{1}{1-A(z)}$

Review: symbolic method to count permutations

How many permutations of length N?

Class	P, the class of all permutations
Size	$\|p\|$, the length of p
OGF	$P(z) = \sum_{p \in P} \frac{z^{\|p\|}}{\|p\|!} = \sum_{N \geq 0} P_N \frac{z^N}{N!}$

Atom

type	*class*	*size*	*GF*
labelled atom	Z	1	z

Construction $$P = E + Z \star P$$

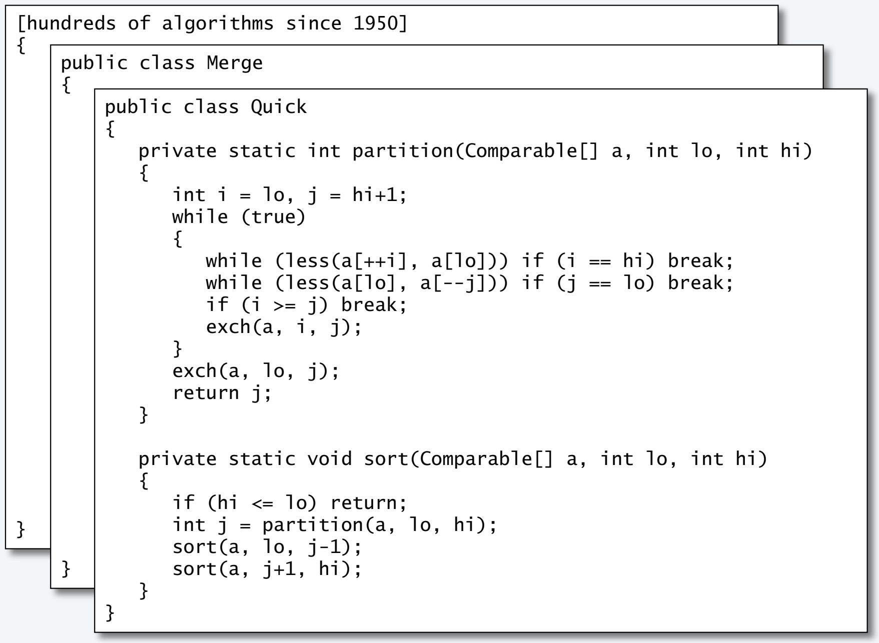
"a permutation is empty or an atom and a permutation"

OGF equation $$P(z) = 1 + zP(z)$$

Solution $$P(z) = \frac{1}{1-z}$$

$$N![z^N]P(z) = N! \ ✓$$

Application: Sorting algorithms



```
[hundreds of algorithms since 1950]
{
   public class Merge
   {
      public class Quick
      {
         private static int partition(Comparable[] a, int lo, int hi)
         {
            int i = lo, j = hi+1;
            while (true)
            {
               while (less(a[++i], a[lo])) if (i == hi) break;
               while (less(a[lo], a[--j])) if (j == lo) break;
               if (i >= j) break;
               exch(a, i, j);
            }
            exch(a, lo, j);
            return j;
         }

         private static void sort(Comparable[] a, int lo, int hi)
         {
}           if (hi <= lo) return;
            int j = partition(a, lo, hi);
   }        sort(a, lo, j-1);
            sort(a, j+1, hi);
         }
      }
```

input (maybe not in random order)

T S R P O N M L I

random permutation of the input

N L T R M O I P S

sorted output

I L M N O P R S T

Q. Model for input?

A. Random permutation.

Q. Realistic?

Q. Absolutely, if we put entries in random order before the sort!

Chapter 2

Application: Randomly permuting an array/generate a random permutation

Algorithm (Knuth)

- Move from left to right.
- Exch each entry with a *random* entry to its right.

```
for (int i = 0; i < N; i++)
{
   int r = i + StdRandom.uniform(N-i);
   int t = a[r]; a[r] = a[i]; a[i] = t;
}
```

All permutations are equally likely:

- 1st entry equally likely to be any of the *N* entries.
- 2nd equally likely to be any of the $N-1$ remaining entries.
- 3rd equally likely to be any of the $N-2$ remaining entries.
- ...

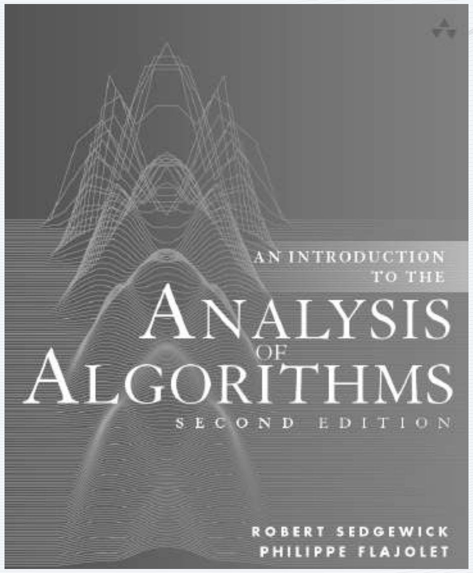
input (maybe not in random order)

T	S	R	P	O	N	M	L	I
N	S	R	P	O	**T**	M	L	I
N	**L**	R	P	O	T	M	**S**	I
N	L	**T**	P	O	**R**	M	S	I
N	L	T	**R**	O	**P**	M	S	I
N	L	T	R	**M**	P	**O**	S	I
N	L	T	R	M	**O**	P	S	I
N	L	T	R	M	O	**I**	S	**P**
N	L	T	R	M	O	I	**P**	**S**

random permutation of the input

N	L	T	R	M	O	I	P	S
6	8	1	3	7	5	9	4	2

use 1 2 3 4 5 6 7 8 9 as input to get a random *permutation* $\longrightarrow$



http://aofa.cs.princeton.edu

7. Permutations

- Basics
- **Sets of cycles**
- Left-right-minima
- Other parameters
- BGFs and distributions

Review: Permutations and derangements

How many sets of cycles of length N?

"A permutation is a set of cycles"

Construction $$P^* = SET(CYC(Z))$$

EGF equation $$P^*(z) = \exp\Big(\ln \frac{1}{1-z}\Big) = \frac{1}{1-z}$$

Counting sequence $$P^*_N = N![z^N]P^*(z) = N!$$

How many derangements of length N?

"Derangements are permutations with no singleton cycles"

Construction $$D = SET(CYC_{>1}(Z))$$

EGF equation $$D(z) = e^{z^2/2+z^3/3+z^4/4+\ldots} = \exp\Big(\ln \frac{1}{1-z} - z\Big) = \frac{e^{-z}}{1-z}$$

Expansion $$[z^N]D(z) \equiv \frac{D_N}{N!} = \sum_{0\le k\le N} \frac{(-1)^k}{k!} \sim \frac{1}{e}$$

Review: generalized derangements

How many permutations of length N have no cycles of length $\leq M$?

Construction $$D_M = SET(CYC_{>M}(Z))$$

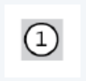
OGF equation $$D_M(z) = e^{\frac{z^{M+1}}{M+1} + \frac{z^{M+2}}{M+2} + \dots} = \exp\Big(\ln\frac{1}{1-z} - z - z^2/2 - \ldots - z^M/M\Big)$$

$$= \frac{e^{-z - \frac{z^2}{2} - \frac{z^3}{3} - \dots \frac{z^M}{M}}}{1-z}$$

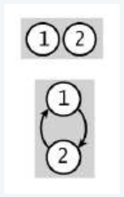
Asymptotics $$[z^N]D_M(z) \sim \frac{N!}{e^{H_M}}$$

Involutions

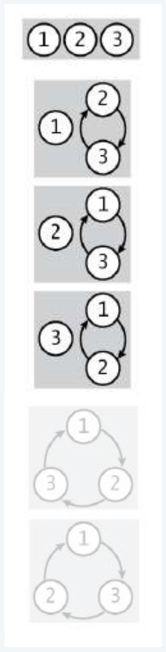
are permutations composed of cycles of length 1 or 2.



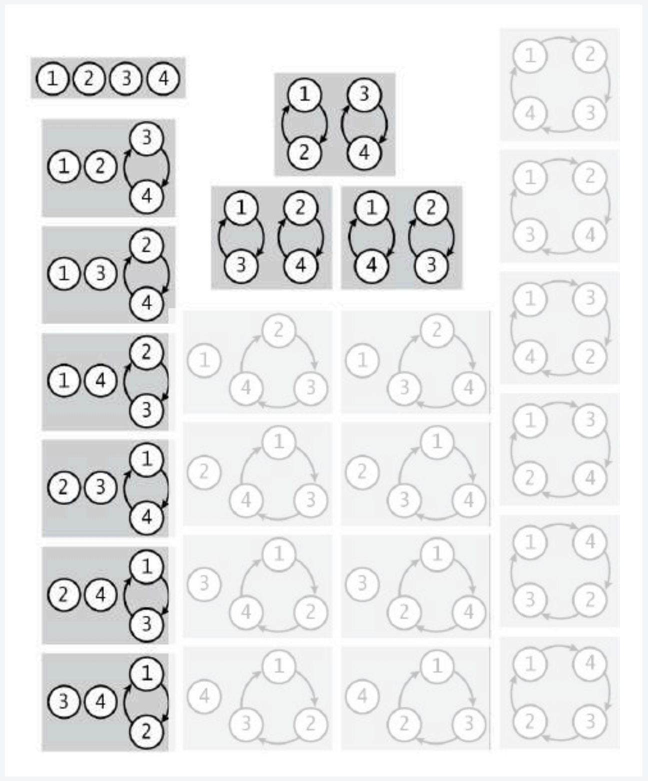
$I_1 = 1$



$I_2 = 2$



$I_3 = 4$



$I_4 = 10$

Review: Inverse

Alternate def. A *permutation* is a mapping of the numbers 1 through N to itself.

index	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
permutation	9	12	11	10	5	15	1	3	7	6	13	8	2	16	4	14

Def. The *inverse* of a permutation is the inverse of that mapping.

7	13	8	15	5	10	9	12	1	4	3	2	11	16	6	14
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

index	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
inverse	7	13	8	15	5	10	9	12	1	4	3	2	11	16	6	14

Q. What is the inverse of an *involution*?

Inverse of an involution

An *involution* is a mapping of the numbers 1 through *N* to itself with all 1- or 2-cycles

index	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
involution	9	12	11	4	5	15	7	13	1	10	3	2	8	16	6	14

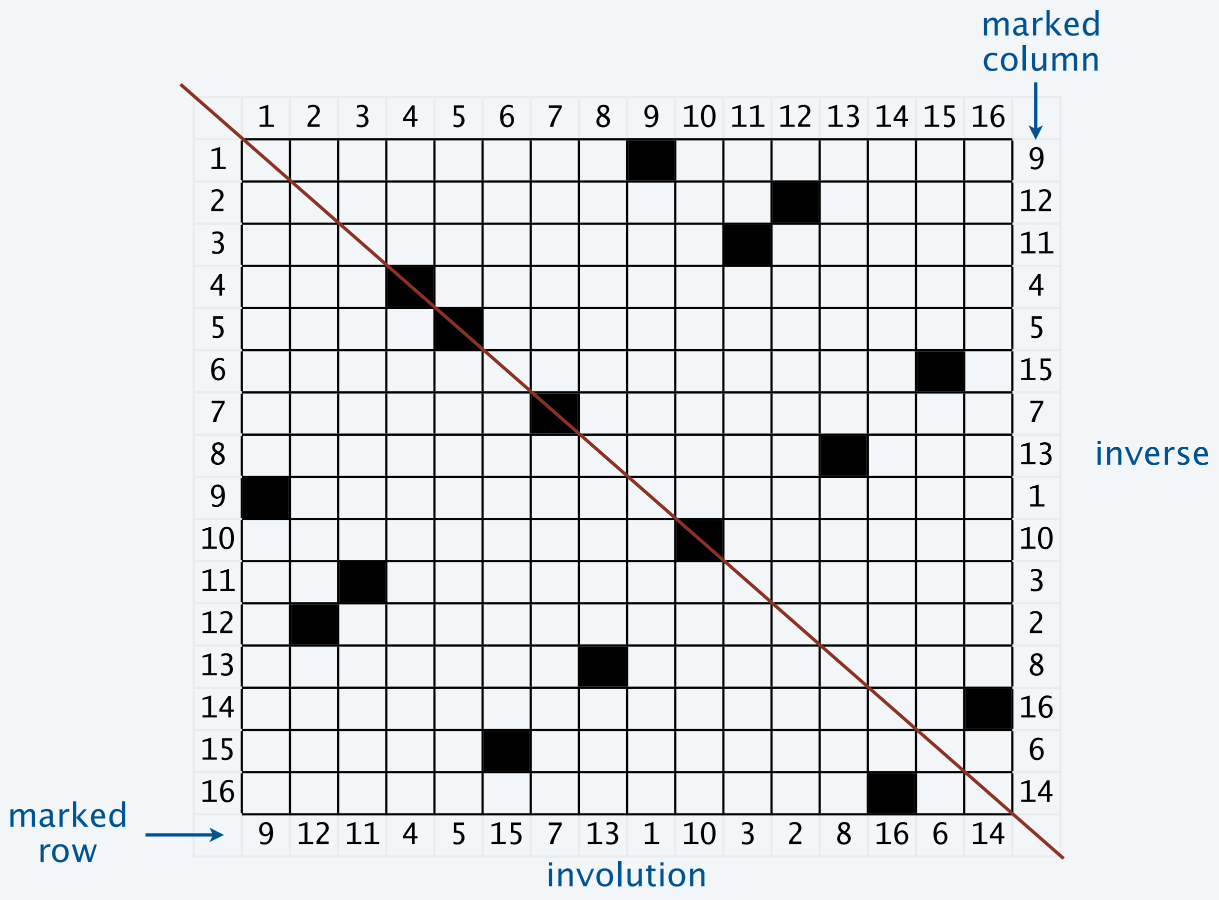
Def. The *inverse* of an involution is the inverse of that mapping.

9	12	11	4	5	15	7	13	1	10	3	2	8	16	6	14
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
inverse	9	12	11	4	5	15	7	13	1	10	3	2	8	16	6	14

Q. What is the inverse of an involution? **A.** ITSELF!

Lattice representation of an involution



Representation of involution is *symmetric* about the main diagonal.

Application: Reciprocal cipher

An *involution* is a permutation that is its own inverse.

Implication: Can encrypt and decrypt with the same machine.

Enigma (WW II)

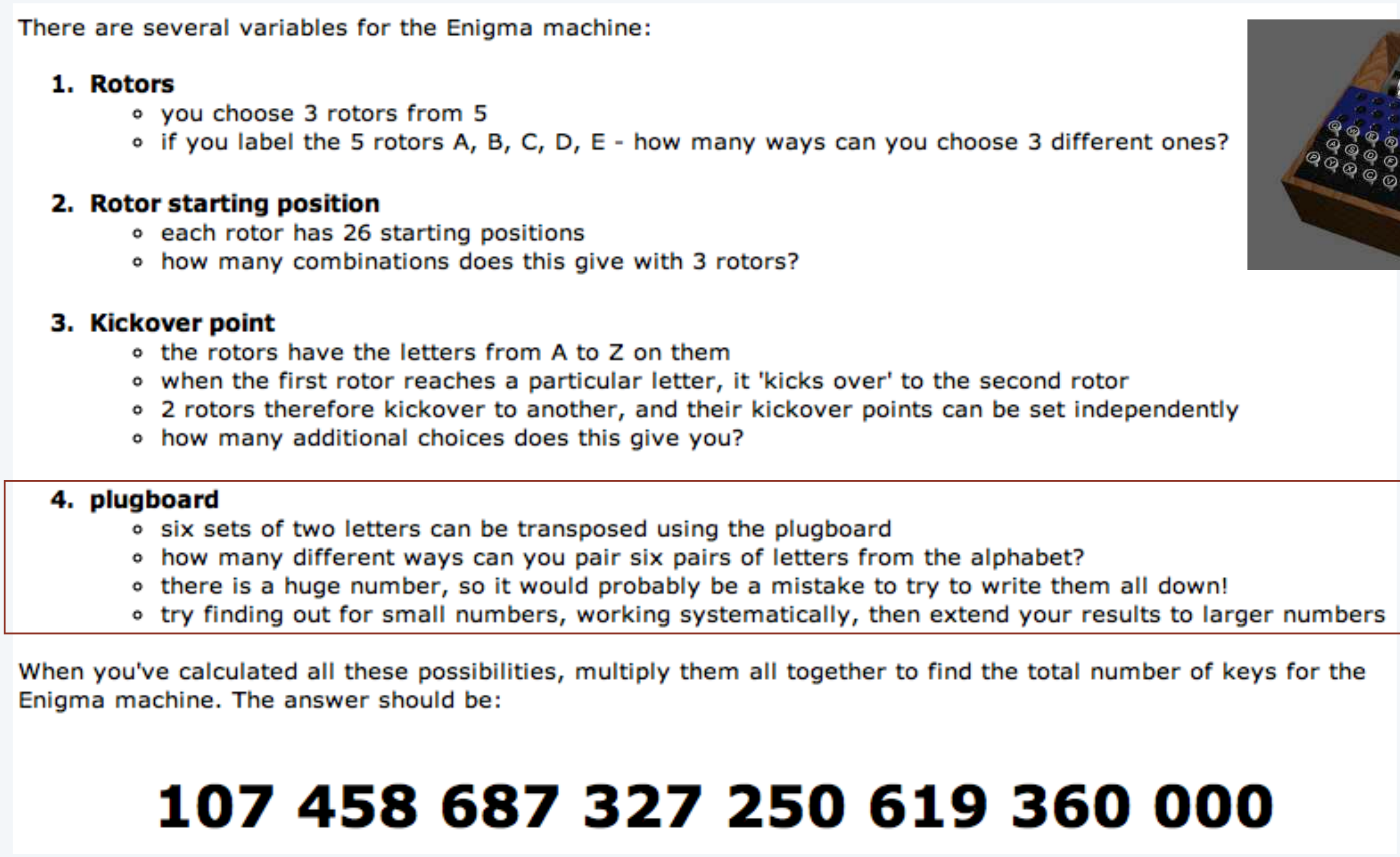
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	-
involution	D	K	R	A	Z	F	U	J	N	J	B	L	X	I	O	S	V	C	P	-	G	Q	Y	M	W	E	T

Encryption	plaintext	A	T	T	A	C	K	-	A	T	-	D	A	W	N
	ciphertext	D	K	R	A	Z	F	U	J	N	J	B	L	X	I

Decryption	ciphertext	D	K	R	A	Z	F	U	J	N	J	B	L	X	I
	plaintext	A	T	T	A	C	K	-	A	T	-	D	A	W	N

Caveat. Still susceptible to character frequency analysis but can be useful as a *component*.

Application: How many different Enigma settings?



There are several variables for the Enigma machine:

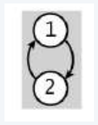
1. **Rotors**
   - you choose 3 rotors from 5
   - if you label the 5 rotors A, B, C, D, E - how many ways can you choose 3 different ones?
2. **Rotor starting position**
   - each rotor has 26 starting positions
   - how many combinations does this give with 3 rotors?
3. **Kickover point**
   - the rotors have the letters from A to Z on them
   - when the first rotor reaches a particular letter, it 'kicks over' to the second rotor
   - 2 rotors therefore kickover to another, and their kickover points can be set independently
   - how many additional choices does this give you?
4. **plugboard**
   - six sets of two letters can be transposed using the plugboard
   - how many different ways can you pair six pairs of letters from the alphabet?
   - there is a huge number, so it would probably be a mistake to try to write them all down!
   - try finding out for small numbers, working systematically, then extend your results to larger numbers

When you've calculated all these possibilities, multiply them all together to find the total number of keys for the Enigma machine. The answer should be:

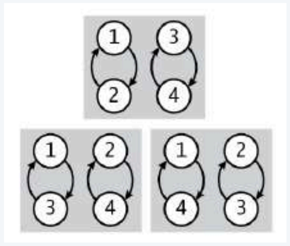
107 458 687 327 250 619 360 000

Warmup

How many perms are comprised entirely of 2-cycles?



$R_2 = 1$



$R_4 = 3$

Example: ROT-13 (world's weakest cryptosystem)

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M

Encryption

```
A T T A C K   A T   D A W N
N G G N P X   N G   Q N J A
```

Decryption

```
N G G N P X   N G   Q N J A
A T T A C K   A T   D A W N
```

Construction $R = SET(CYC_2(Z))$

OGF equation $R(z) = e^{z^2/2}$

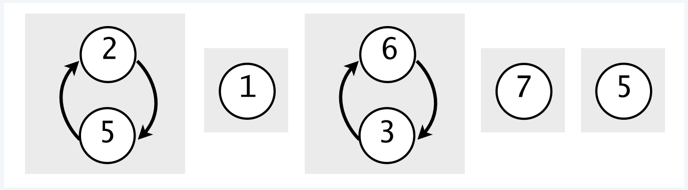
Coefficients $R_N \equiv N![z^N]e^{z^2/2} \dfrac{N!}{2^{N/2}(N/2)!} \sim \sqrt{2}\Big(\dfrac{N}{4e}\Big)^{N/2}$

Stirling's approximation

$N! \sim (N/e)^N\sqrt{2\pi N}$

Involutions

How many involutions of size N?



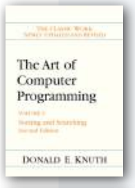
Construction $$I = SET(CYC_1(Z)) \star SET(CYC_2(Z))$$

"Involutions are permutations with all cycles of length 1 or 2"

OGF equation $$I(z) = e^{z + z^2/2}$$

Coefficients $$I_N \equiv N![z^N]e^{z+z^2/2} = \sum_{0 \le 2k \le N} \frac{N!}{k!2^k(N-2k)!}$$

Laplace method



Asymptotics $$\sim \frac{1}{\sqrt{2\sqrt{e}}} \left(\frac{N}{e}\right)^{N/2} e^{\sqrt{N}}$$

Complex asymptotics
(stay tuned for Part 2)

Generalized involutions

How many permutations of length N have no cycles of length > M ?

Construction $$I_M = SET(CYC_1(Z)) \star SET(CYC_2(Z)) \star \ldots \star SET(CYC_M(Z))$$

OGF equation $$I_M(z) = e^{z + z^2/2 + \ldots + z^M/M}$$

Coefficient asymptotics $$I_{MN} \sim \frac{1}{\sqrt{2\pi\lambda}} \frac{e^{1+r/2+\ldots+r^M/M}}{r^N}$$

Complex asymptotics (stay tuned for Part 2)

In-class exercise

Find $[z^{10}]e^{z + z^2/2 + z^3/3 + z^4/4 + z^5/5}$

$$= [z^{10}]e^{\ln\frac{1}{1-z} - z^6/6 - z^7/7 - z^8/8 - z^9/9 - z^{10}/10 - \ldots}$$

$$= [z^{10}]\frac{1}{1-z}e^{-z^6/6}e^{-z^7/7}e^{-z^8/8}e^{-z^9/9}e^{-z^{10}/10}\ldots$$

$$= [z^{10}]\frac{1}{1-z}(1-\frac{z^6}{6})(1-\frac{z^7}{7})(1-\frac{z^8}{8})(1-\frac{z^9}{9})(1-\frac{z^{10}}{10})\ldots$$

$$= [z^{10}](1+z+z^2+\ldots+z^{10})\big(1-\frac{z^6}{6}-\frac{z^7}{7}-\frac{z^8}{8}-\frac{z^9}{9}-\frac{z^{10}}{10}\big)$$

$$= 1-\frac{1}{6}-\frac{1}{7}-\frac{1}{8}-\frac{1}{9}-\frac{1}{10} \doteq 35438$$

100 prisoners

Problem. 100 prisoners, each uniquely identified by a numbered ID card (1 to 100), have been sentenced to death, but are given a last chance.

- The ID cards are collected and put in the drawers of a cabinet with 100 numbered drawers (1 to 100) in random order, one card per drawer
- One at a time, the prisoners are allowed to enter the room containing the cabinet and open, then close again, at most *half* the drawers.
- If *all* prisoners find their own number, they will all be spared.
- If *one prisoner fails*, they will all be executed.

Prisoner A, a mathematician, bemoans their fate, claiming the probability of success is on the order of $2^{-100} \approx 8 \cdot 10^{-31}$.

Prisoner B, who knows analytic combinatorics, claims to know a strategy that gives them better than 30% chance of success.

What is Prisoner B's strategy?

100 prisoners solution

Problem. 100 prisoners, each uniquely identified by a numbered ID card (1 to 100), have been sentenced to death, but are given a last chance.

- The ID cards are collected and put in the drawers of a cabinet with 100 numbered drawers (1 to 100) in random order, one card per drawer.
- One at a time, the prisoners are allowed to enter the room containing the cabinet and open, then close again, at most *half* the drawers.
- If *all* prisoners find their own number, they will all be spared.
- If *one prisoner fails*, they will all be executed.

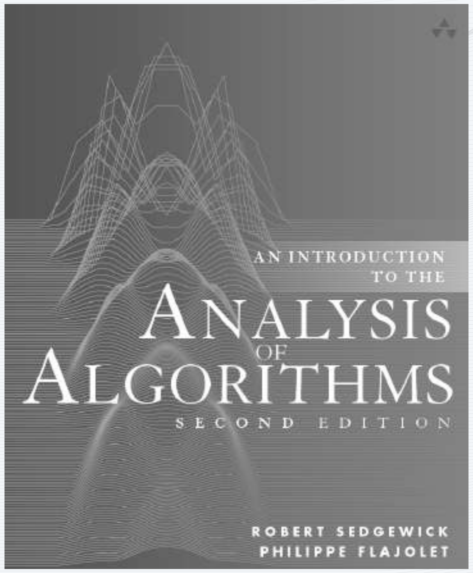
Prisoner B's strategy: Each prisoner "follows the cycle"

- Opens the drawer corresponding to his ID.
- Uses the number in that drawer to decide which drawer to open next.
- Continues until finding the drawer containing his ID.

Q. When does Prisoner's B strategy succeed?

A. When the random permutation has no cycles of length greater than 50.

Probability of success: $[z^{100}]\exp\left(\frac{z}{1}+\frac{z}{2}+\ldots+\frac{z}{50}\right) = 1-(H_{100}-H_{50}) \doteq 0.31$



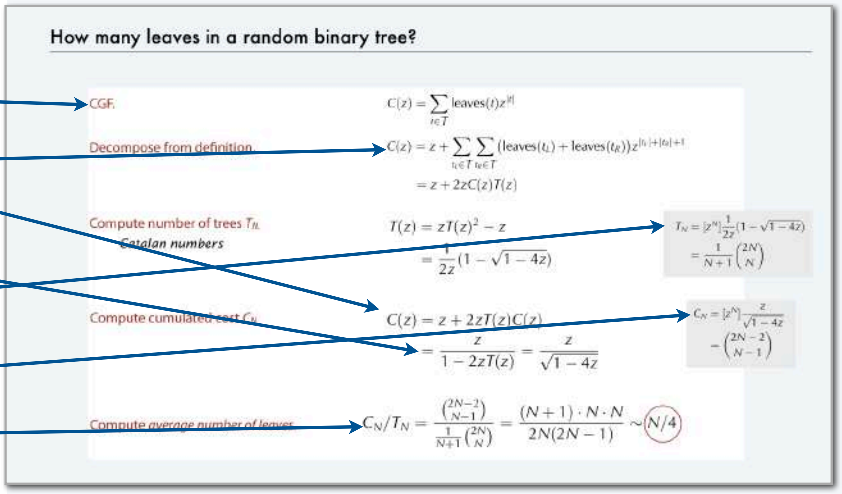
http://aofa.cs.princeton.edu

7. Permutations

General approach for analyzing parameters

Review: Cumulated cost approach for parameters

- Define GF for counting sequence and CGF.
- Identify construction to give CGF equation.
- Solve to get explicit formula for CGF.
- Extract coefficients from GF to get counting seq.
- Extract coefficients from CGF to get cumulated cost.
- Divide to compute expected value



How many leaves in a random binary tree?

CGF. $C(z) = \sum_{t \in T} \text{leaves}(t) z^{|t|}$

Decompose from definition. $C(z) = z + \sum_{t_L \in T} \sum_{t_R \in T} (\text{leaves}(t_L) + \text{leaves}(t_R)) z^{|t_L|+|t_R|+1}$
$= z + 2zC(z)T(z)$

Compute number of trees T_N. $T(z) = zT(z)^2 - z$
$= \frac{1}{2z}(1 - \sqrt{1-4z})$

Catalan numbers

$T_N = [z^N]\frac{1}{2z}(1-\sqrt{1-4z})$
$= \frac{1}{N+1}\binom{2N}{N}$

Compute cumulated cost C_N. $C(z) = z + 2zT(z)C(z)$
$= \frac{z}{1-2zT(z)} = \frac{z}{\sqrt{1-4z}}$

$C_N = [z^N]\frac{z}{\sqrt{1-4z}}$
$= \binom{2N-2}{N-1}$

Compute *average number of leaves* $C_N/T_N = \frac{\binom{2N-2}{N-1}}{\frac{1}{N+1}\binom{2N}{N}} = \frac{(N+1)\cdot N \cdot N}{2N(2N-1)} \sim N/4$

Small trick for permutations:

- Use *exponential* CGF.
- Treat as OGF to extract expected value directly.

Why does it work?

- $N!$ is the normalizing factor for ECGF.
- $N!$ is *also* the counting sequence.

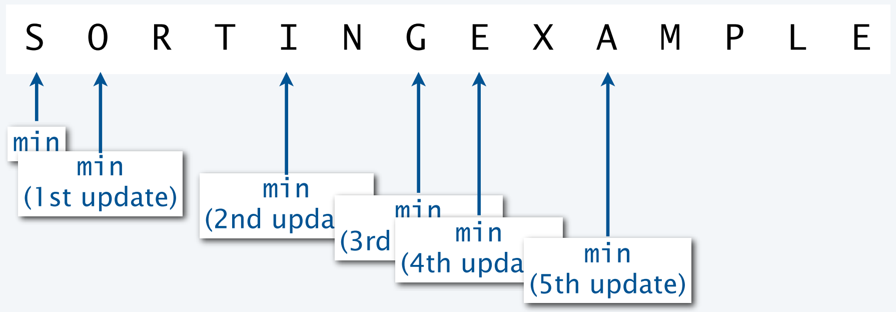
$$B(z) = \sum_{p \in \mathcal{P}} \text{cost}(p) \frac{z^{|p|}}{|p|!} = \sum_{N \geq 0} B_N \frac{z^N}{N!}$$

cumulated cost $\longrightarrow$ / counting sequence $\longrightarrow$

$$\frac{N![z^N]B(z)}{N!} = [z^N]B(z) = \frac{B_N}{N!}$$

Application: Selection sort

```
public static void sort(Comparable[] a)
{
   int N = a.length;
   for (int i = 0; i < N; i++)
   {
      int min = i;
      for (int j = i+1; j < N; j++)
         if (less(a[j], a[min])) min = j;
      exch(a, i, min);
   }
}
```



Section 2.1

Q. How many times is min updated in the first pass (assuming keys distinct)?

A. The number of left-right minima in the permutation.

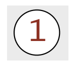
Q. How many left-right minima in a random permutation?

Caveat. Cost for whole sort is complicated, but not significant relative to the number of compares.

Left-right minima

Def. A left-right minimum (lrm) in a permutation is a smaller than any item to its left.

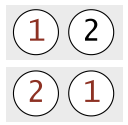
Q. How many *lrm* in a random permutation of size N?



$P_1 = 1$

$B_1 = 1$

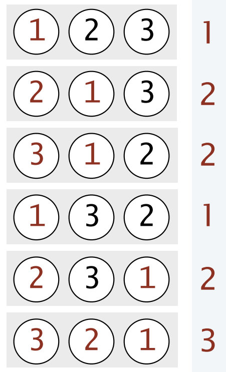
$B_1/P_1 = 1$



$P_2 = 2$

$B_2 = 1 + 2 = 3$

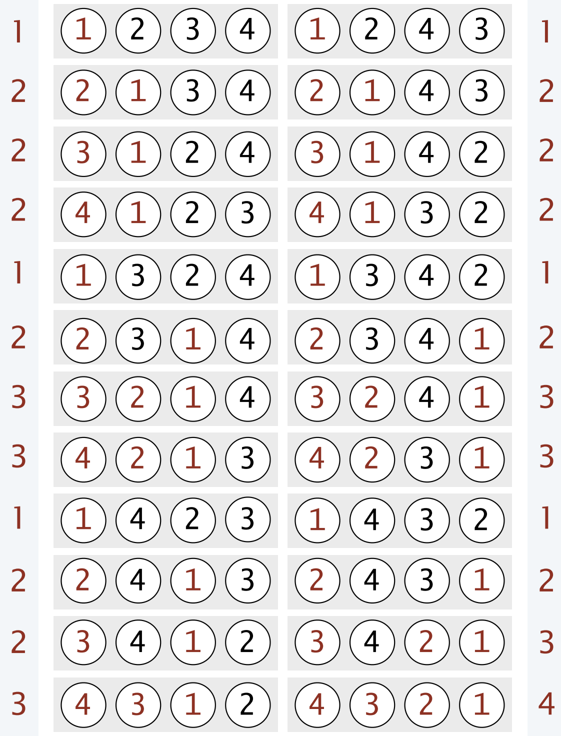
$B_2/P_2 = 3/2 = 1.5$



$P_3 = 6$

$B_3 = 2\cdot 1 + 3\cdot 2 + 1\cdot 3 = 11$

$B_3/P_3 = 11/6 \doteq 1.833$



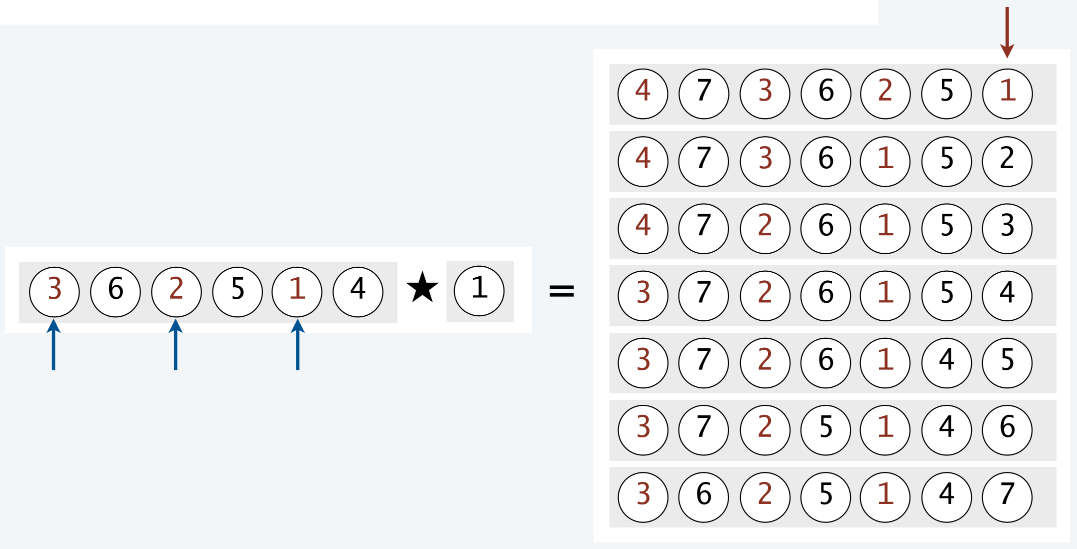
$P_4 = 24$

$B_4 = 6\cdot 1 + 11\cdot 2 + 6\cdot 3 + 1\cdot 4 = 50$

$B_4/P_4 = 50/24 \doteq 2.083$

Construction for left-right minima

Create |p|+1 perms from a perm p by star product construction.



Original perm has lrm(p) left-right minima.

Q. How many left-right minima in the set of constructed perms?

A. $(|p| + 1)\ \mathrm{lrm}(p) \quad + 1$

$|p|$ + 1 copies of the original perm

only the one ending in 1 adds a lrm

Average number of left-right minima in a random permutation

CGF.
$$B(z) = \sum_{p\in\mathcal{P}} \text{lrm}(p)\frac{z^{|p|}}{|p|!} \qquad = \sum_{N\geq 0} B_N\frac{z^N}{N!}$$

Apply construction.
$$= \sum_{p\in\mathcal{P}} \big((|p|+1)\text{lrm}(p)+1\big)\frac{z^{|p|+1}}{(|p|+1)!}$$

Simplify.
$$= \sum_{p\in\mathcal{P}} \text{lrm}(p)\frac{z^{|p|+1}}{(|p|)!} + \sum_{p\in\mathcal{P}} \frac{z^{|p|+1}}{(|p|+1)!}$$

Substitute.
$$= zB(z) + \sum_{k\geq 0}\frac{z^{k+1}}{(k+1)} \quad = zB(z) + \ln\frac{1}{1-z}$$

Solve.
$$B(z) = \frac{1}{1-z}\ln\frac{1}{1-z}$$

OGF for the Harmonic numbers

Expand.
$$[z^N]B(z) = \frac{B_N}{N!} = \boxed{H_N}$$

cumulated cost

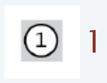
average # lrm in a random permutation

$$H_1 = 1$$
$$H_2 = 1 + \frac{1}{2} = 1.5$$
$$H_3 = 1 + \frac{1}{2} + \frac{1}{3} \doteq 1.833$$
$$H_4 = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} \doteq 2.083$$

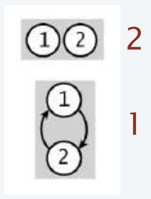
✓

Cycles

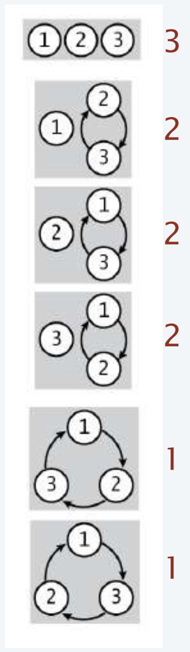
Q. How many *cycles* in a random permutation of size N?



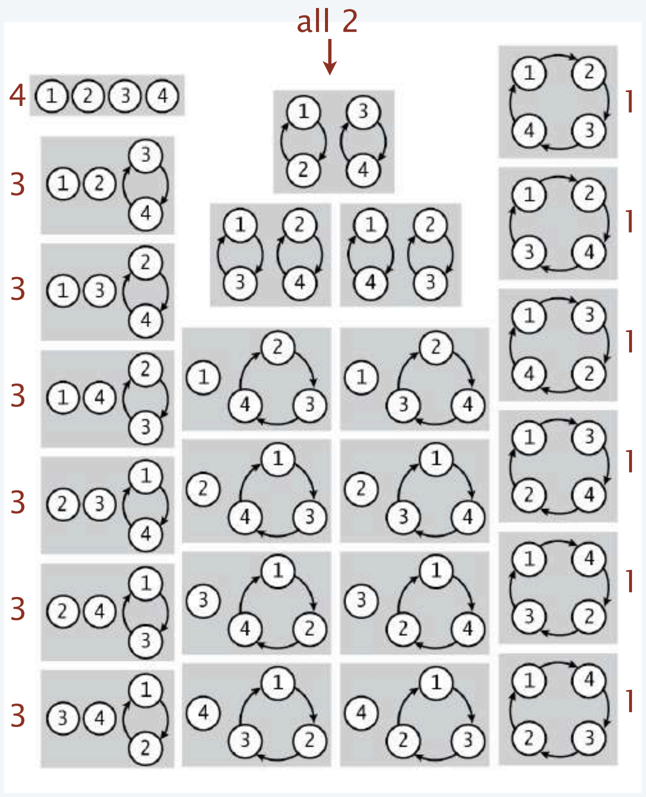
$B_1 = 1$
$B_1/P_1 = 1$



$B_2 = 1 + 2 = 3$
$B_2/P_2 = 3/2 = 1.5$



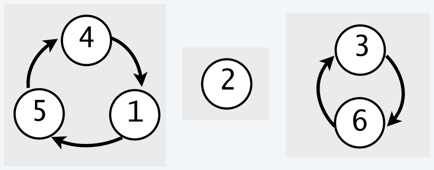
$B_3 = 2{\cdot}1 + 3{\cdot}2 + 1{\cdot}3 = 11$
$B_3/P_3 = 11/6 \doteq 1.833$

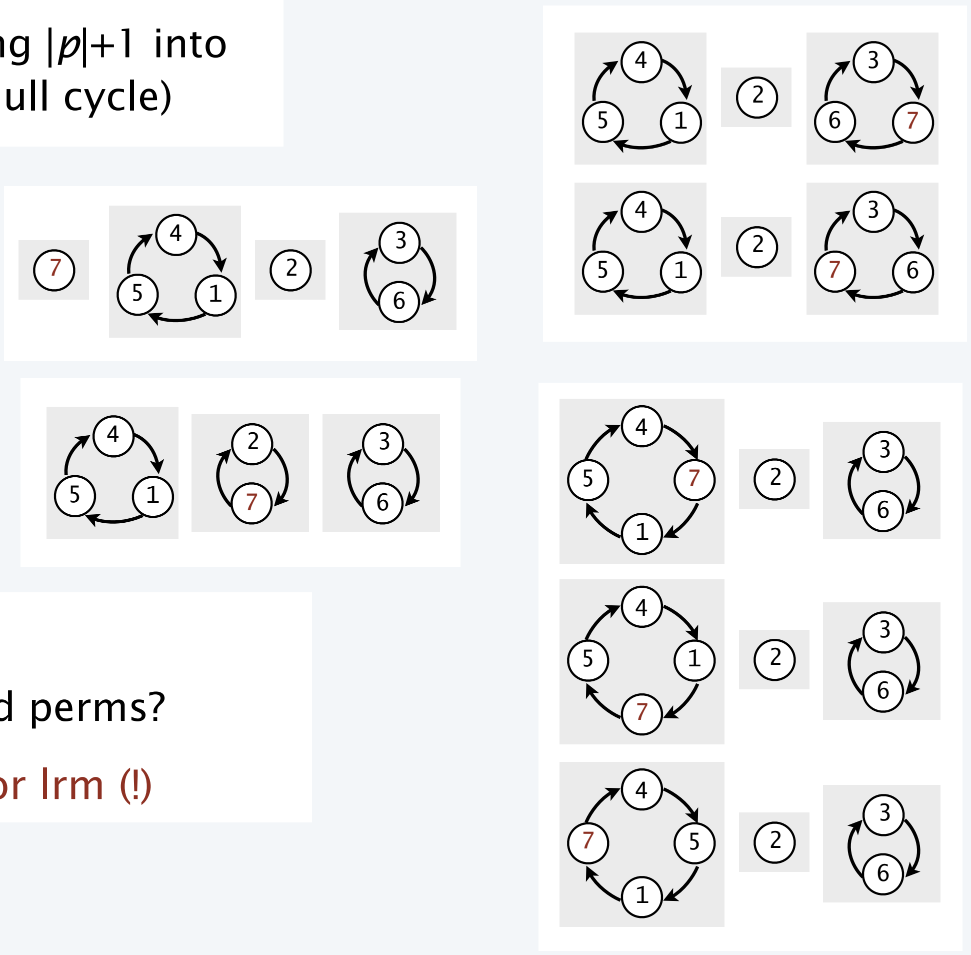


$B_4 = 6{\cdot}1 + 11{\cdot}2 + 6{\cdot}3 + 1{\cdot}4 = 50$
$B_4/P_4 = 50/24 \doteq 2.083$

Construction for cycles

Create $|p|+1$ perms from a perm p by inserting $|p|+1$ into every position in every cycle (including the null cycle)





Original perm has cycles(p) cycles.

Q. How many cycles in the set of constructed perms?

A. $\underbrace{(|p| + 1)\ \text{cycles}(p)}_{|p|\ +\ 1\ \text{copies of the original perm}} \quad \underbrace{+\ 1}_{\text{from the null cycle}}$ ⟵ same as for lrm (!)

Average number of cycles in a random permutation (same derivation as for lrm)

CGF.
$$B(z) = \sum_{p\in\mathcal{P}} \text{cycles}(p)\frac{z^{|p|}}{|p|!} = \sum_{N\geq 0} B_N \frac{z^N}{N!}$$

Decompose.
$$= \sum_{p\in\mathcal{P}} \big((|p|+1)\text{cycles}(p)+1\big)\frac{z^{|p|+1}}{(|p|+1)!}$$

Simplify.
$$= \sum_{p\in\mathcal{P}} \text{cycles}(p)\frac{z^{|p|+1}}{(|p|)!} + \sum_{p\in\mathcal{P}} \frac{z^{|p|+1}}{(|p|+1)!}$$

Substitute.
$$= zB(z) + \sum_{k\geq 0}\frac{z^{k+1}}{(k+1)} = zB(z) + \ln\frac{1}{1-z}$$

Solve.
$$B(z) = \frac{1}{1-z}\ln\frac{1}{1-z}$$

OGF for the Harmonic numbers

Expand.
$$[z^N]B(z) = \underbrace{\frac{B_N}{N!}}_{\text{cumulated cost}} = \boxed{H_N}$$

cumulated cost

average # cycles in a random permutation

$$H_1 = 1$$
$$H_2 = 1 + \frac{1}{2} = 1.5$$
$$H_3 = 1 + \frac{1}{2} + \frac{1}{3} \doteq 1.833 \quad ✓$$
$$H_4 = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} \doteq 2.083$$

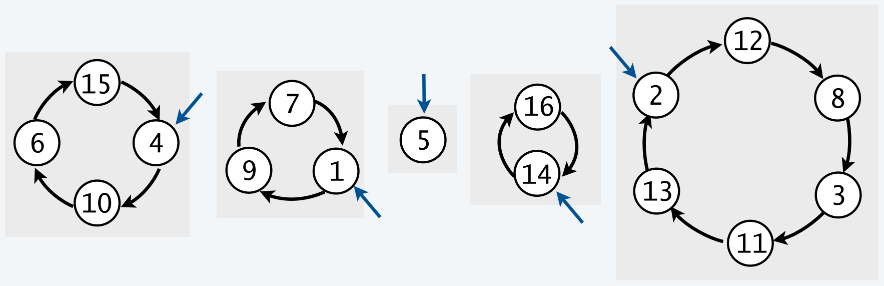
Left-right minima and cycles

Q. Is there a 1:1 correspondence?

A. Yes!

To build a permutation from a set of cycles:

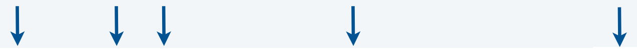
- Identify smallest as the *leader* in each cycle.
- Write cycles in *decreasing* order of leader.



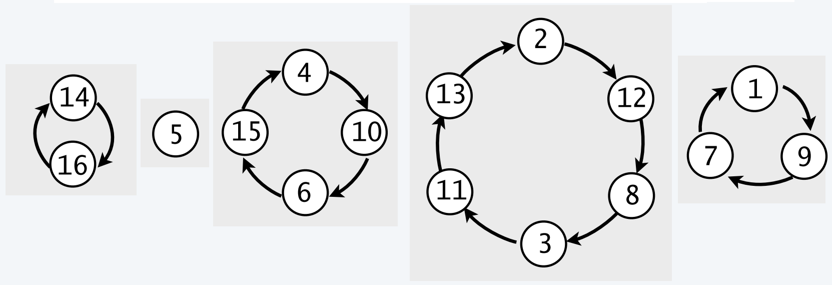
14 16 5 4 10 6 15 2 12 8 3 11 13 1 7 9

To build a set of cycles from a permutation:

- Identify left-right minima.
- Build cycles with entries delimited by lrms (start a new cycle with each lrm).



14 16 5 4 10 6 15 2 12 8 3 11 13 1 7 9


14
16
5
4
10
6
15
2
12
8
3
11
13
1
9
7