Building a Bombe

Jonah Weinbaum

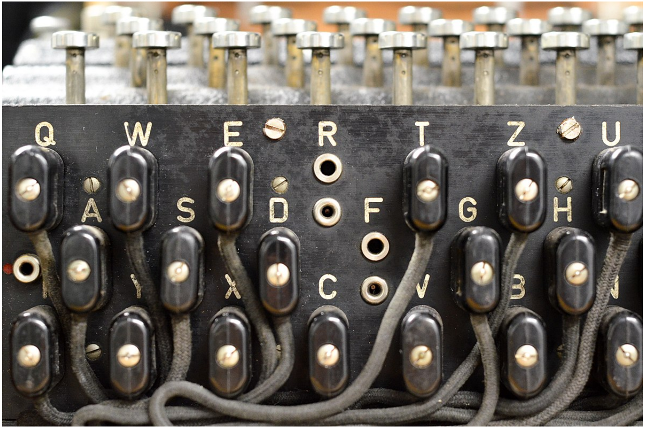
Computer Science Theory Seminar

September 20, 2024

The Enigma Machine

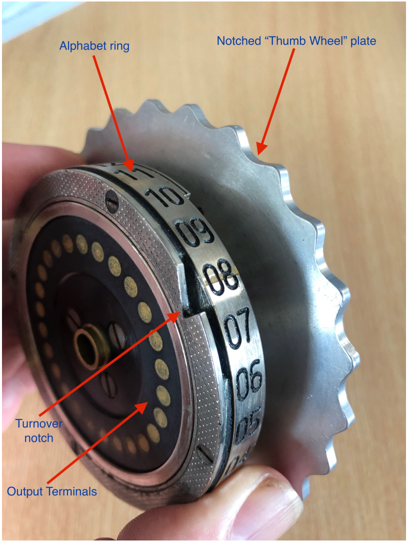
The Enigma Machine

The Enigma Machine

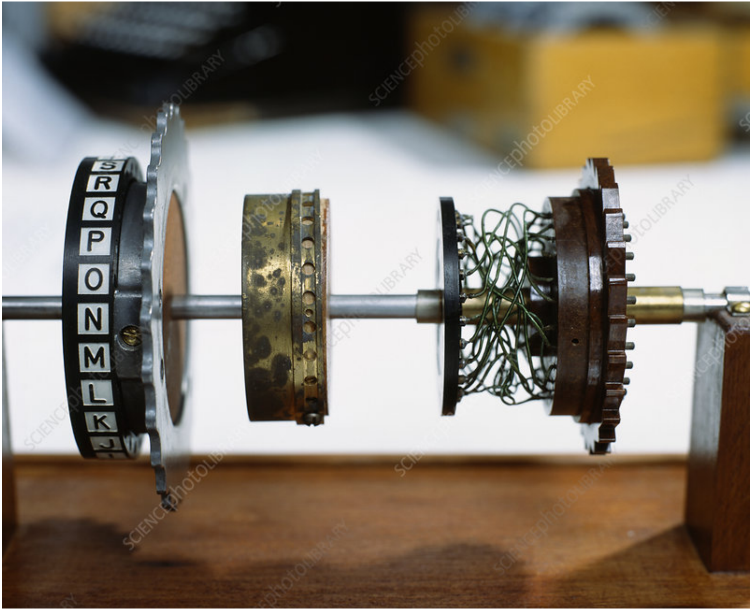


The Enigma Machine

The Enigma Machine



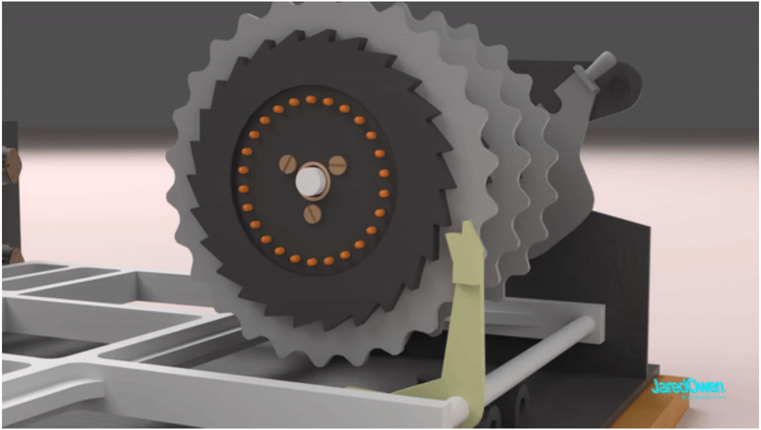
The Enigma Machine



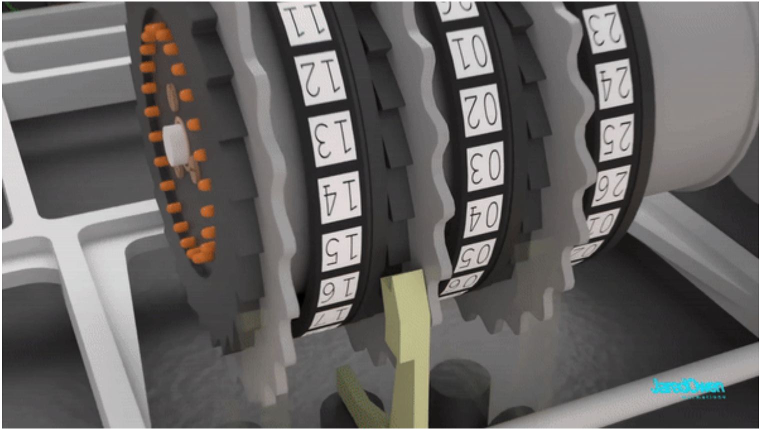
The Enigma Machine

The Enigma Machine

The Enigma Machine



The Enigma Machine



Encryption Space (M3 Naval Enigma)

- 5 rotors to choose from
- 26 rotor positions
- 26 ring settings
- 10 plugboard wires
- 2 reflectors (*very rarely 3*)

Encryption Space (M3 Naval Enigma)

- $\binom{5}{3} \cdot 3!$
- 26 rotor positions
- 26 ring settings
- 10 plugboard wires
- 2 reflectors (*very rarely 3*)

Encryption Space (M3 Naval Enigma)

- $\binom{5}{3} \cdot 3!$
- 26^3
- 26 ring settings
- 10 plugboard wires
- 2 reflectors (*very rarely 3*)

Encryption Space (M3 Naval Enigma)

- $\binom{5}{3} \cdot 3!$
- 26^3
- 26^2
- 10 plugboard wires
- 2 reflectors (*very rarely 3*)

Plugboard (Order Matters)

$$\begin{aligned}
&\binom{26}{2}\binom{24}{2}\cdots\binom{8}{2} \\
&= \frac{26!}{2 \cdot 24!} \cdot \frac{24!}{2 \cdot 22!} \cdots \frac{8!}{2 \cdot 6!} \\
&= \frac{26!}{2^{10} \cdot 6!}
\end{aligned}$$

Plugboard (Order Doesn't Matter)

$$\frac{26!}{2^{10} \cdot 6!} \cdot \frac{1}{10!} = 150738274937250$$

Encryption Space (M3 Naval Enigma)

- $\binom{5}{3} \cdot 3!$
- 26^3
- 26^2
- $\frac{26!}{2^{10} \cdot 6!} \cdot \frac{1}{10!}$
- 2

Encryption Space (M3 Naval Enigma)

$$2.14917374654501238 72 \cdot 10^{23} \approx 2^{77}$$

A **77-bit key space** in an era before computers!

Permutations

Definition

A permutation on a set S is a bijective function $\sigma : S \to S$

Example

- The identity permutation $\mathrm{id}(x) = x \ \forall\ x \in S$
- Caesar ciphers

Permutations

The set of permutations on a set S is denoted $\mathrm{Sym}(X)$ and forms a group under the operation of function composition.

- The identity is id_S
- Inverse is the function inverse of a permutations

Example

$S_n := \mathrm{Sym}(\mathbb{N}_n)$

Permutations

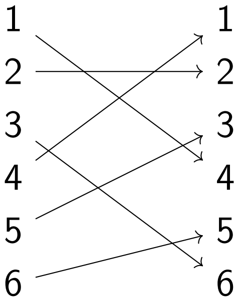
The Caesar cipher is one of the simplest encryption schemes. It involves shifting the set of letters by a fixed amount to encode a message. For example, $A \mapsto D, \ldots, X \mapsto A, Y \mapsto B, Z \mapsto C$. In the context of permutations, this can be viewed as a repeated application of the Caesar permutation by one letters θ_1. For instance, to get Caesar's particular cipher we use $\theta_1 \circ \theta_1 \circ \theta_1$ (that is $\theta_1^3 \in \mathrm{Sym}(\{A, \ldots, Z\})$). For ease of notation we define

$$\theta_n := \theta_1^n \text{ for } n \in \mathbb{N}$$

Permutations

Repeated applications of a permutation on a finite group must eventually return to a previously found value. This is known as a cycle of a permutation.

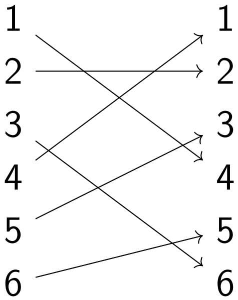
Permutations



Has cycles

- $1 \mapsto 4 \mapsto 1$
- $2 \mapsto 2$
- $3 \mapsto 6 \mapsto 5 \mapsto 3$

Permutations



We write this as

$$(14)(2)(365)$$

Permutations

All permutations can be decomposed in this way, and doing so gives us a unique (up to order) **cycle type** of the permutation

Example

$(14)(2)(365)$ has cycle type $(2, 1, 3)$

Enigma as a Permutation

Supposing we are at ground position, the Enigma permutation E is as follows:

$$E = P^{-1}\alpha_1^{-1}\alpha_2^{-1}\alpha_3^{-1}R\alpha_3\alpha_2\alpha_1 P$$

Enigma as a Permutation

After n moves of the first rotor this is

$$E_n = P^{-1}\theta_n^{-1}\alpha_1^{-1}\theta_n^{-1}\alpha_2^{-1}\alpha_3^{-1}R\alpha_3\alpha_2\theta_n^{-1}\alpha_1\theta_n P$$

Where θ_n is a shift by n letters.

Enigma as a Permutation

$$\begin{aligned} E &= P^{-1}\alpha_1^{-1}\alpha_2^{-1}\alpha_3^{-1}R\alpha_3\alpha_2\alpha_1 P \\ &= (\alpha_3\alpha_2\alpha_1 P)^{-1}R(\alpha_3\alpha_2\alpha_1 P) \end{aligned}$$

Enigma as a Permutation

Conjugate permutations will always have the same cycle type

Enigma as a Permutation

An Enigma machine will *always* be a permutation represented by a 13 disjoint 2 cycles

Sending Messages

Geheim! **Sonder - Maschinenschlüssel BGT**

Datum	Walzenlage	Ringstellung	Steckerverbindungen	Grundstellung
31.	IV II I	F T R	HR AT IW SK UY DF GV LJ BO MX	vyj
30.	III V II	Y V P	OR KI JV QE ZN MU BF YC DS GP	cqr
29.	V IV I	O H R	UX JC PB BK TA ED ST DS LU FI	vhf

When sending a message the operator was to use the following protocol

I. The operator sets their machine to the ground position specified by the key sheet
II. A *Spruchschlüssel* rotor position is chosen and encoded twice using the *Grundstellung*, this is listed
III. The message is encoded using the daily key and the *Spruchschlüssel* rotor position

Demonstration

Suppose we receive a message with starting letters

```
ABC DEF
```

Given that the Spruchschlüssel was encoded twice at ground position we know A and D, represent the same letter in the Spruchschlüssel.

Key Distribution Vulnerability

Suppose the Spruchschlüssel is $\alpha\beta\gamma$ Then the recieved message can be interpreted as

$$\begin{gathered}\texttt{ABC DEF}\\ E_1(\alpha)E_2(\beta)E_3(\gamma)\ E_4(\alpha)E_5(\beta)E_6(\gamma)\end{gathered}$$

Then $E_4(E_1(A)) = E_4(\alpha) = D$

Key Distribution Vulnerability

With sufficient messages we can completely deduce $E_4E_1, E_5E_2,$ and E_6E_3. There three cycle types serve a *footprint* for the initial setting E. By preparing a catalogue of these cycle types for each ground position, we drastically reduce our search space.

The Cyclometer

Changes to Protocol

Starting in 1940, the German's enhanced the security of their key distribution. Originally, the *Grundstellung* rotor position was sent along with the daily key and an operator chose a *Spruchschlusse* to encode twice at the start of a message. Later iterations of this protocol removed the *Grundstellung* from key sheets entirely.

Changes to Protocol

These new key sheets contained the following columns *Tag/Datum*, *Walzenlage*, *Ringstellung*, *Steckerverbindungen*, and *Kenngruppen*

Changes to Protocol

Geheime Kommandosache!

Nicht ins Flugzeug mitnehmen

Armee-Stabs-Maschinenschlüssel Nr. 28

für Oktober 1944

Nr. 00008

	Datum	Walzenlage			Ringstellung			Steckerverbindungen										Kenngruppen			
St	31.	IV	V	I	21	15	16	KL	IT	FQ	HY	XC	NP	VZ	JB	SE	OG	jkm	ogi	ncj	glp
St	30.	IV	II	III	26	14	11	ZN	YO	QB	ER	DK	XU	GP	TV	SJ	LM	ino	udl	nam	lax
St	29.	II	V	IV	19	09	24	ZU	HL	CQ	WM	OA	PY	EB	TR	DN	VI	nci	oid	yhp	nip
St	28.	IV	III	I	03	04	22	YT	BX	CV	ZN	UD	IR	SJ	HW	GA	KQ	zqj	hlg	xky	ebt
St	27.	V	I	IV	20	06	18	KX	GJ	EP	AC	TB	HL	MW	QS	DV	OZ	bvo	sur	ccc	lqe
St	26.	IV	I	V	10	17	01	YV	GT	OQ	WN	FI	SK	LD	RP	MZ	BU	jhx	uuh	giw	ugw
St	25.	V	IV	III	13	04	17	QR	GB	HA	NM	VS	WD	YZ	OF	XK	PE	tba	pnc	ukd	nld
St	24.	III	II	IV	09	20	18	RS	NC	WK	GO	YQ	AX	EH	VJ	ZL	PF	nfi	mew	xbk	yes
St	23.	V	II	III	11	21	08	EY	DT	KF	MO	XP	HN	WG	ZL	IV	JA	lsd	nuo	vcr	vcx
St	22.	I	II	IV	01	25	02	PZ	SE	OJ	XF	HA	GB	VQ	UY	KW	LR	yji	rwy	rdk	nso
St	21.	IV	I	III	06	22	03	GH	JR	TQ	KF	NZ	IL	WM	BD	UO	EC	ema	mlv	jjy	iqh
St	20.	V	I	II	12	25	08	TF	RQ	XV	DZ	PY	NL	WI	SJ	ME	GB	xjl	pgs	ggh	znd
St	19.	IV	III	II	07	05	23	ZX	EU	AC	GD	KP	VO	QS	NW	HL	RM	vpj	zqe	jrs	cgm
St	18.	II	III	V	19	14	22	WG	OM	RL	~~DB~~	~~ST~~	~~AQ~~	~~PZ~~	~~XH~~	~~YN~~	~~IJ~~	~~oxd~~	~~lnb~~	~~ieu~~	~~ytt~~
St	17.	IV	I	II	12	08	21	ME	HX	BF	WY	ZD	TR	FJ	AG	IL	KQ	tak	pjs	kdh	jvh
St	16.	I	II	III	07	11	15	WZ	AB	MO	TF	RX	SG	QU	VJ	YN	EL	pzg	evw	wyt	iye
St	15.	III	II	V	06	16	02	GT	YC	EJ	LA	RX	PN	IS	WB	MH	ZV	bhe	xzm	yzk	evp
St	14.	II	I	V	23	05	24	AZ	CJ	WF	UY	SO	QV	MI	NH	DP	GX	fdx	tyj	bmq	typ
St	13.	IV	II	V	03	25	10	CX	KN	JR	DQ	IU	TL	HZ	MF	EP	WB	zfo	bjr	zwx	gvn
St	12.	I	III	II	26	01	18	QB	YE	WN	AI	GJ	TO	HR	FK	PS	CM	upc	anf	tkr	pwz
St	11.	V	I	III	17	13	04	SV	GO	PA	ZR	FN	HI	YM	WT	DE	BJ	vdh	ego	wmy	uti
St	10.	I	V	IV	26	07	16	SW	AQ	NP	FO	VY	UX	MK	CL	HT	ZJ	rpl	anw	vpr	mhn
St	9.	I	III	IV	17	10	18	EH	IR	GK	NZ	SP	UA	LD	CQ	JM	YV	knq	ysq	rhj	tlj
St	8.	V	II	I	23	11	25	QY	OG	ST	HA	CB	WD	KL	JN	VX	IU	lro	avw	axh	gws
St	7.	II	III	I	06	12	03	BG	FS	TH	JE	VK	PI	CU	QA	OD	NM	aty	mbb	mvo	jmz
St	6.	I	IV	V	24	19	01	IR	HQ	NT	WZ	VC	OY	GP	LF	BX	AK	bhc	iwo	zgz	rnr
St	5.	II	IV	III	05	22	14	MK	GO	RQ	XT	DW	IA	ZL	SY	PJ	EN	bok	rzw	kzo	ryl
St	4.	IV	II	I	15	02	21	KD	PG	CO	FW	HJ	RY	MT	QL	VB	UZ	kpk	php	xmo	pfw
St	3.	III	V	IV	03	23	04	DY	CP	WN	OV	QH	UZ	RA	TI	GL	SM	hjy	nkt	ytn	pvc
St	2.	I	III	V	13	18	01	DR	VJ	FS	ZK	IU	HX	AQ	GT	YO	FC	gpq	fqw	oiy	ruj
St	1.	II	IV	I	06	17	26	AC	LS	BQ	WN	MY	UV	FJ	PZ	TR	OK	ool	ooi	ywv	sfb

When sending a message the operator was to use the following protocol

I. The time at which the message was sent is listed

II. The number of parts which the message contained is listed

III. Which message part is being sent is listed

IV. The length of the message part (not including *Buchstabenkenngruppe*) is listed

V. A *Grundstellung* rotor position is chosen and listed

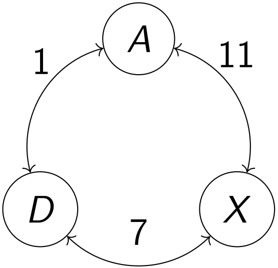
VI. A *Spruchschlüssel* rotor position is chosen and encoded using the *Grundstellung*, this is listed

VII. The *Buchstabenkenngruppe* is listed

VIII. The message part encoded using the daily key and the *Spruchschlüssel* position is listed

Suppose we knew the plaintext which had been enciphered into a particular Enigma transmission. Consider the following mapping,

$$\begin{tikzcd}
D \arrow[d, "1"'] & Y \arrow[d, "2"'] & Y \arrow[d, "3"'] & Y \arrow[d, "4"'] & Y \arrow[d, "5"'] & Y \arrow[d, "6"'] & X \arrow[d, "7"'] & Y \arrow[d, "8"'] & Y \arrow[d, "9"'] & Y \arrow[d, "10"'] & X \arrow[d, "11"'] \\
A & B & R & A & C & A & D & A & B & R & A
\end{tikzcd}$$


A
1
11
D
7
X

We denote the permutation represented by the Enigma at position i as σ_i. Since these each use the same plugboard we will also note the Enigma at position i not using the plugboard as $\overline{\sigma_i}$, that is $\sigma_i = P\overline{\sigma_i}P$ (conversely, $\overline{\sigma}_i = P\sigma_i P$).

$$\begin{aligned}\sigma_{11} \circ \sigma_7 \circ \sigma_1 &= P\overline{\sigma_{11}}P \circ P\overline{\sigma_7}P \circ P\overline{\sigma_1}P \\ &= P \circ \overline{\sigma_{11}} \circ \overline{\sigma_7} \circ \overline{\sigma_1} \circ P\end{aligned}$$

We will condense this notation by defining

$$\sigma \coloneqq \sigma_{11} \circ \sigma_7 \circ \sigma_1$$

and

$$\overline{\sigma} \coloneqq \overline{\sigma_{11}} \circ \overline{\sigma_7} \circ \overline{\sigma_1}$$

Then $\sigma = P\overline{\sigma}P$ (conversely, $\overline{\sigma} = P\sigma P$).

Let us hypothesize that A is steckered in the plugboard to α – that is, $P(A) = \alpha$ (conversely, $P(\alpha) = A$). It then follows that for a fixed $i \in \mathbb{N}$

$$\begin{aligned}\overline{\sigma}^i(\alpha) &= P \circ \sigma^i \circ P(\alpha) \\ &= P \circ \sigma^i(A) \\ &= P(A)\end{aligned}$$

and so we derive

$$P(A) = \alpha \Rightarrow P(A) = \overline{\sigma}^i(\alpha) \ \forall \ i \in \mathbb{N}$$

Then we have that A must be steckered to all values in the set $\{\overline{\sigma}^i(\alpha) \mid i \in \mathbb{N}\}$. We note that this set is that orbit of the element α under the group action of the subgroup $\langle\overline{\sigma}\rangle$ – that is, $\langle\overline{\sigma}\rangle \cdot \alpha$.

By representing $\overline{\sigma}$ in its cycle notation we can quickly see whether certain hypotheses are possible. For example, suppose we found that

$$\overline{\sigma} = (ABCDEF)(GHIJK)(L)(MNOPQRSTUVWXYZ)$$

If we suppose that A is steckered to any element in the cycle $(ABCDEF)$ we find that this element has an orbit of length 6 in $\langle \overline{\sigma} \rangle$ and thus A cannot be steckered to any element in this cycle. Then it is clear that A can only be steckered to L in this case.

Scanning Methods

Turing describes various methods of mechanising the above analysis of cycle-type to determine when we can eliminate rotor positions.

Scanning Methods

If we examine a particular hypothesis, say A is steckered to K, we can rule out this steckering if we find that K is not in a 1-cycle, that is if $\overline{\sigma}(K) \neq K$. If we mechanize this process we can eliminate rotor positions which do not satisfy this singular hypothesis. Turing called this method **single line scanning**. Note, however, that this method may eliminate rotor positions which do have valid steckerings, just not the particular steckering that we hypothesized.

Scanning Methods

If we perform single line scanning in sequence, that is, for each steckering hypothesis, we can rule out rotor positions which have all steckering hypotheses invalid. Turing called this method **serial scanning**.

Scanning Methods

Serial scanning requires a separate examination of each steckering. Turing proposed a machine which could concurrently examine all steckering possibilities and eliminate rotor positions which had no valid steckerings. Turing called this method **simultaneous scanning**.

Scanning Methods

If we find $\overline{\sigma}$ has a 26-cycle, then we must have that there are no 1-cycles and thus no valid steckerings. It then follows that the rotor position is incorrect. If we mechanism this process we can eliminate *some* rotor positions which do not have valid steckerings. We will call this method **spider scanning**. Note, however, that this method would not, for example, detect that a $13, 13$-cycle contains no valid steckerings.

Scanning Methods

Turing explained, "The ideal machine that Welchman was aiming at was to reject any position in which a certain fixed-for-the-time Stecker hypothesis led to any direct contradiction... The spider does more than this in one way and less in another. It is not restricted to dealing with one Stecker hypothesis at a time, and it does not find all direct contradictions."

Effectively, spider-scanning is like a form of simultaneous scanning which is restricted to examining only one cycle at a time.

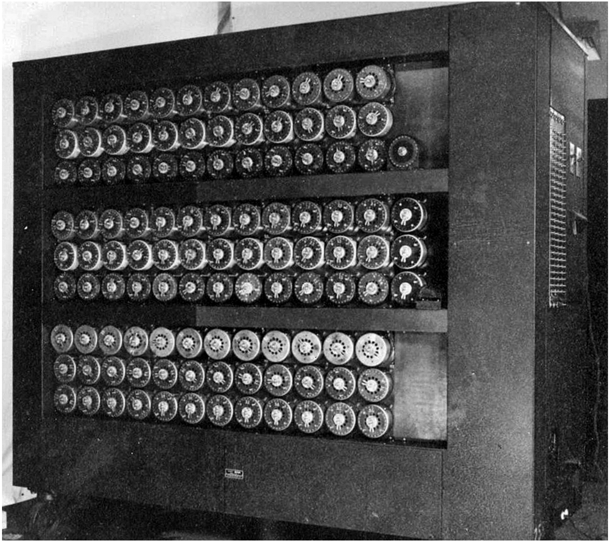
Scanning Methods

Iterations of each scanning methods were proposed or designed, but in the end we find that the spider scanning method was used in the implementation of the **Bombe**.

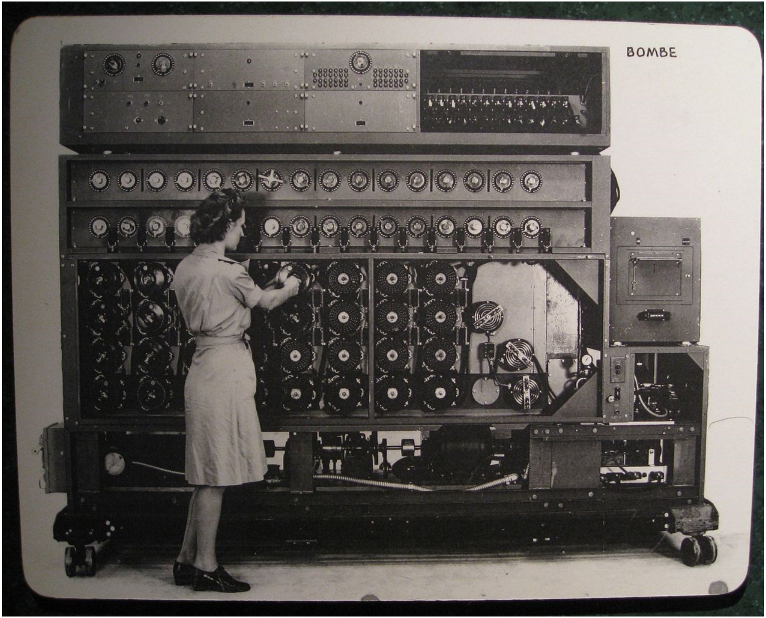
Cribs

```
U A E N F V R L B Z P W M E P M I H F S R J X F M J K W R A X Q E Z
K E I N E B E S O N D E R E N E R E I G N I S S E
0

U A E N F V R L B Z P W M E P M I H F S R J X F M J K W R A X Q E Z
  K E I N E B E S O N D E R E N E R E I G N I S S E
  1

U A E N F V R L B Z P W M E P M I H F S R J X F M J K W R A X Q E Z
    K E I N E B E S O N D E R E N E R E I G N I S S E
    2

U A E N F V R L B Z P W M E P M I H F S R J X F M J K W R A X Q E Z
      K E I N E B E S O N D E R E N E R E I G N I S S E
      3

U A E N F V R L B Z P W M E P M I H F S R J X F M J K W R A X Q E Z
        K E I N E B E S O N D E R E N E R E I G N I S S E
        4

U A E N F V R L B Z P W M E P M I H F S R J X F M J K W R A X Q E Z
          K E I N E B E S O N D E R E N E R E I G N I S S E
          5

U A E N F V R L B Z P W M E P M I H F S R J X F M J K W R A X Q E Z
            K E I N E B E S O N D E R E N E R E I G N I S S E
            6
```

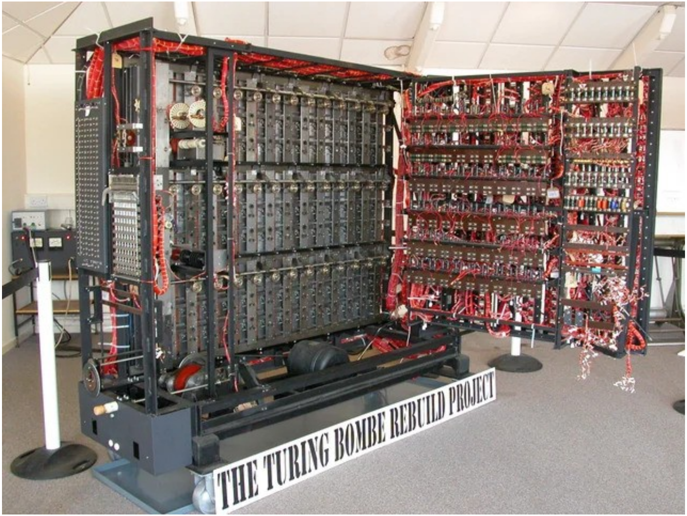
The Bombe



The Bombe

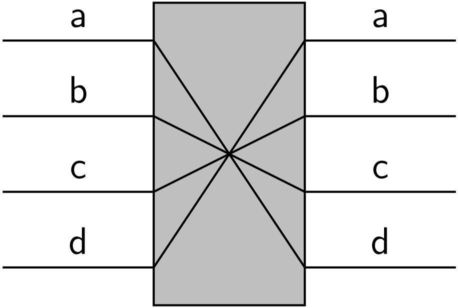


The Bombe



The Bombe

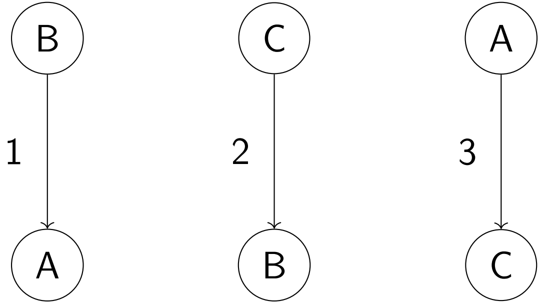
We abstract an Enigma machine



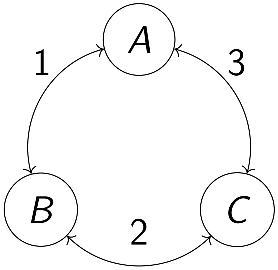
$\overline{\sigma}_\alpha$

The Bombe

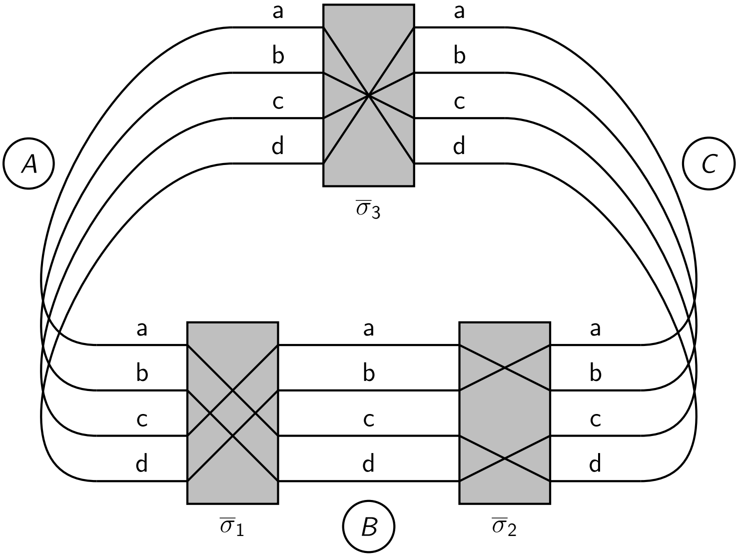
Suppose we had plaintext ciphertext pairing



The Bombe

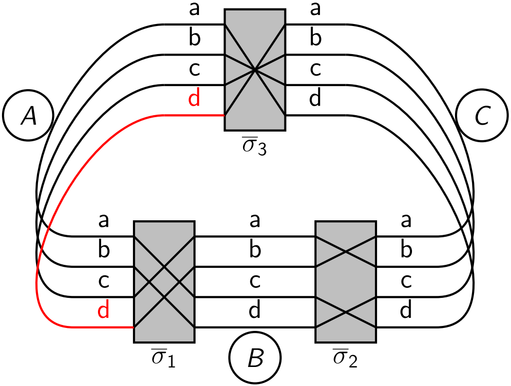


The Bombe



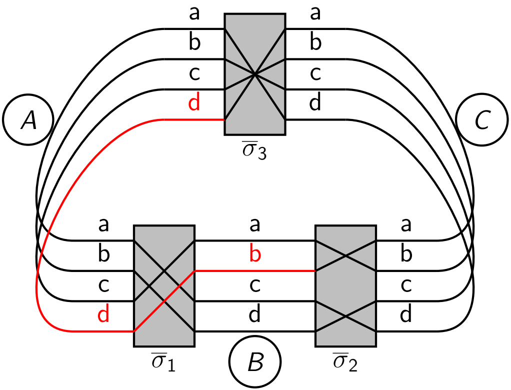
a
b
c
d
a
b
c
d
A
C
$\overline{\sigma}_3$
a
b
c
d
a
b
c
d
a
b
c
d
$\overline{\sigma}_1$
B
$\overline{\sigma}_2$

The Bombe



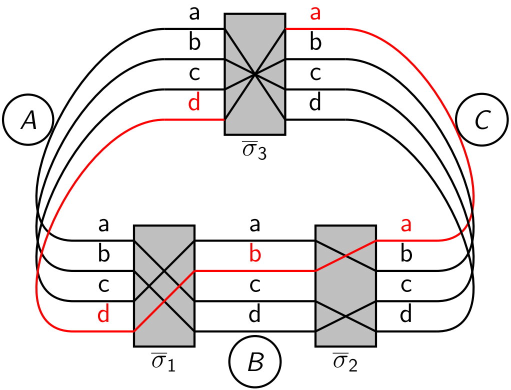
a
b
c
d
a
b
c
d
A
C
$\overline{\sigma}_3$
a
b
c
d
a
b
c
d
a
b
c
d
$\overline{\sigma}_1$
B
$\overline{\sigma}_2$

The Bombe

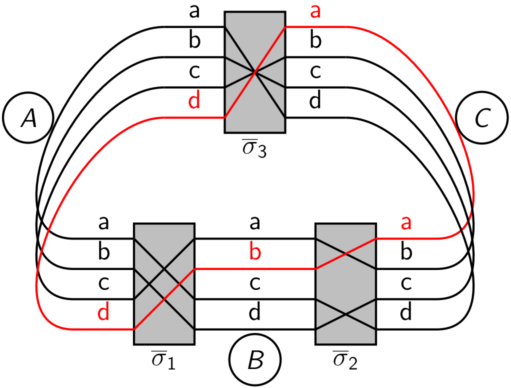


a
b
c
d
a
b
c
d
A
C
$\overline{\sigma}_3$
a
b
c
d
a
b
c
d
a
b
c
d
$\overline{\sigma}_1$
B
$\overline{\sigma}_2$

The Bombe


a
b
c
d
a
b
c
d
$\overline{\sigma}_3$
A
C
a
b
c
d
a
b
c
d
a
b
c
d
$\overline{\sigma}_1$
B
$\overline{\sigma}_2$

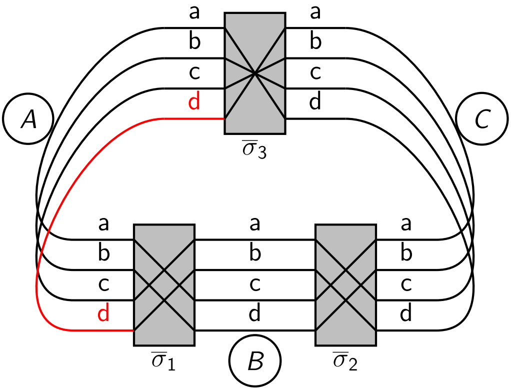
The Bombe



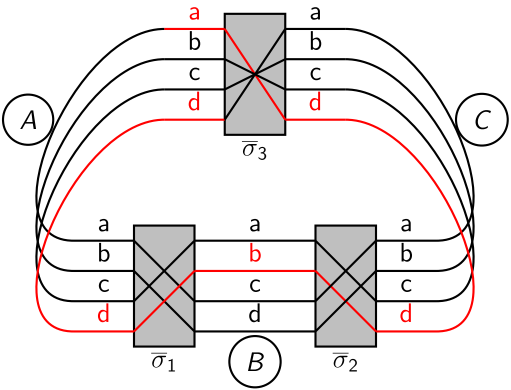
a
b
c
d
a
b
c
d
A
C
$\overline{\sigma}_3$
a
b
c
d
a
b
c
d
a
b
c
d
$\overline{\sigma}_1$
B
$\overline{\sigma}_2$

The Bombe

If we change $\overline{\sigma}$

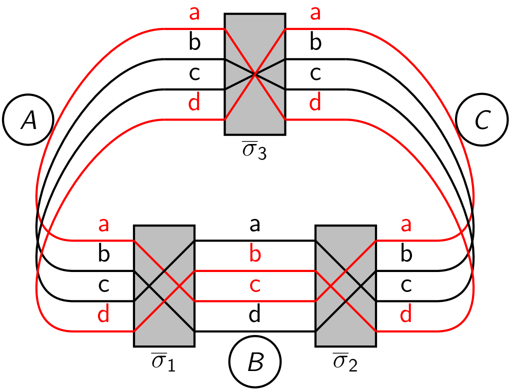


The Bombe



a
b
c
d
a
b
c
d
$\overline{\sigma}_3$
A
C
a
b
c
d
a
b
c
d
a
b
c
d
$\overline{\sigma}_1$
B
$\overline{\sigma}_2$

The Bombe



a
b
c
d
a
b
c
d
$\overline{\sigma}_3$
A
C
a
b
c
d
a
b
c
d
a
b
c
d
$\overline{\sigma}_1$
B
$\overline{\sigma}_2$

The Bombe

Later Welchman introduced the **diagonal board** which made use of the fact that

$$P(X) = Y \iff P(Y) = X$$

To allow us to connect wires Xy and Yx

The Bombe

The **Machine Gun** further improved the device by eliminating stops which had the property that Xy and Vy were both live

Turover Consideration

The Bombe assumes no turnover occurs during encryption. For a crib of n lettters there is an $\frac{n}{26}$ chance of a turnover so we can examine parts of the crib separately to improve the chance that we examine a section with no turnover.

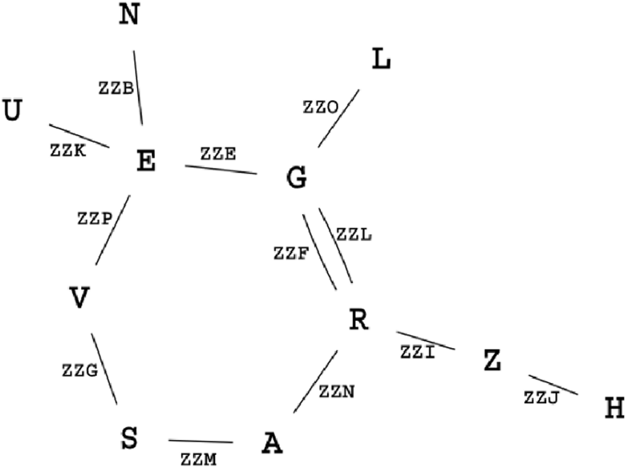
The Bombe

Demonstration

The Bombe

Letter Number		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Cipher		S	N	M	K	G	G	S	T	Z	Z	U	G	A	R	L	V
Clear		W	E	T	T	E	R	V	O	R	H	E	R	S	A	G	E

The Bombe



The Bombe

1: ZZK	U: 1 in
2: ZZE	E: (1 out, 2 in), (7 out, 8 in)
3: ZZF	G: (2 out, 3 in), (11 out, 12 in), input
4: ZZN	R: (3 out, 4 in), (10 out, 11 in)
5: ZZM	A: (4 out, 5 in)
6: ZZG	S: (5 out, 6 in)
7: ZZP	V: (6 out, 7 in)
8: ZZB	N: 8 out
9: ZZJ	H: 9 in
10: ZZI	Z: (9 out, 10 in)
11: ZZL	L: 12 out
12: ZZO	

Current entry at A.

Stops

The machine is wired to stop when $\overline{\sigma}$ has cycle type other than (26). Turing only considers what he calls **normal stops** during his calculation of the expected number of stops. This is a stop which has cycle-type $(25, 1)$.

Stops

Consider our simple example of a loop of three Enigmas on four letters. We might expect that $\overline{\sigma} = \overline{\sigma}_3\overline{\sigma}_2\overline{\sigma}_1$ being generated from considerably random permutations, is itself a random permutation. If this is the case then we would expect that we would get a (4) cycle with a probability of $\frac{1}{4}$. Then we expect the machine to stop with probability $\frac{3}{4}$. With enough loops this probability decreases exponentially and the machine has a tractible number of stops.

Prior Work

The meth od of construction of the table is very tedious and uninteresting. ~~Itxis~~ The table is reproduced below

(H-M for Holland-Martin, of British Tabulating Machine Company)

No. of letters on web	H-M factor
2	0.92
3	0.79
4	0.62
5	0.44
6	0.29
7	0.17
8	0.087
9	0.041
10	0.016
11	0.0060
12	0.0018
13	0.00045
14	0.000095
15	0.000016
16	0.0000023

No. of answers $= 26^{4-c} \times$ H-M factor

c is number of closures

Prior Work

Table 1. H-M factor.

No. of letters in menu	Turing	Recursive star	Computer linear	V
2	0.92	0.9262	0.9262	0.9260
3	0.79	0.7886	0.7886	0.7906
4	0.62	0.6142	0.6142	0.6193
5	0.44	0.4352	0.4350	0.4427
6	0.29	0.2787	0.2785	0.2871
7	0.17	0.1603	0.1600	0.1678
8	0.087	0.08214	0.08190	0.08777
9	0.041	0.03720	0.03701	0.04073
10	0.016	0.01476	0.01463	0.01661
11	0.0060	0.005074	0.005004	0.005897
12	0.0018	0.001496	0.001464	0.001800
13	0.00045	0.0003739	0.0003612	0.0004662
14	0.000095	0.00007817	0.00007411	0.0001009
15	0.000016	0.00001349	0.00001244	0.00001794
16	0.0000023	0.000001895	0.000001676	0.000002570

Stops

However, try as we may, we can never find a collection of Enigma permutations $\{\overline{\sigma}_1, \overline{\sigma}_2, \overline{\sigma}_3\}$ which generate a (4) cycle in $\overline{\sigma}$. This is to say, in our above arrangment, the machine will stop at *every* rotor position thus making the process of checking stops intractable.

Stops

To see why this is the case, note that each $\overline{\sigma}_i$ has cycle type $(2, 2)$ thus they are permutations of even parity. On the other hand, any (4) cycle will have odd parity. When we compose 3 even permutations (i.e. $\overline{\sigma}_3\overline{\sigma}_2\overline{\sigma}_1$) we will always get an even parity permutation, thus this resulting permutation can *never* be a (4) cycle.

Stops

In the case of the Bombe, a cycle of even length can never produce a permutation with a (26) cycle. We can emperically observe this by simulating the Bombe's operation on a cycle of length 8 and we find that every single rotor position produces a stop.

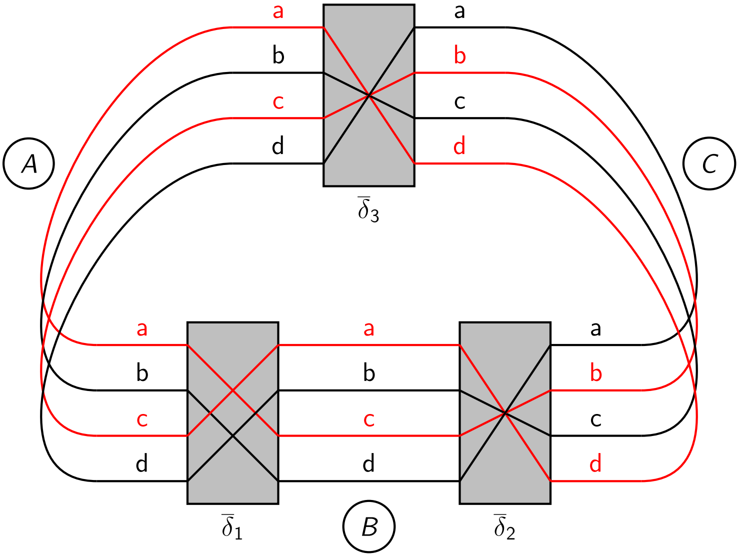
Stops

From the above it is clear that $\overline{\sigma}$ is certainly not a purely random permutation, and simulations of loops of Enigma permutations of various lengths show that the probability distribution of these permutations is highly dependent on the length of the loop.

Stops (Loop Length 3)

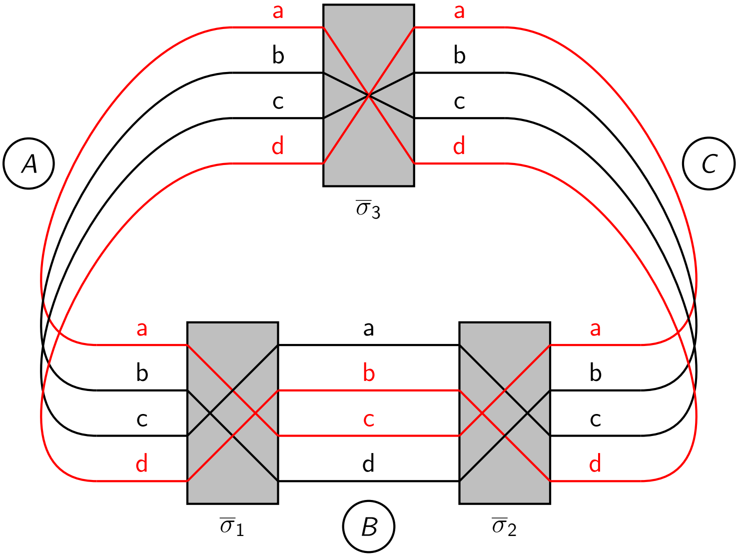
	Probability
(26,)	0.078000
(8, 9, 9)	0.001700
(8, 8, 10)	0.001700
(7, 9, 10)	0.003300
(7, 8, 11)	0.002600
(7, 7, 12)	0.002200
(6, 10, 10)	0.001800
(6, 9, 11)	0.003300
(6, 8, 12)	0.003500
(6, 7, 13)	0.004200
...	

Stops



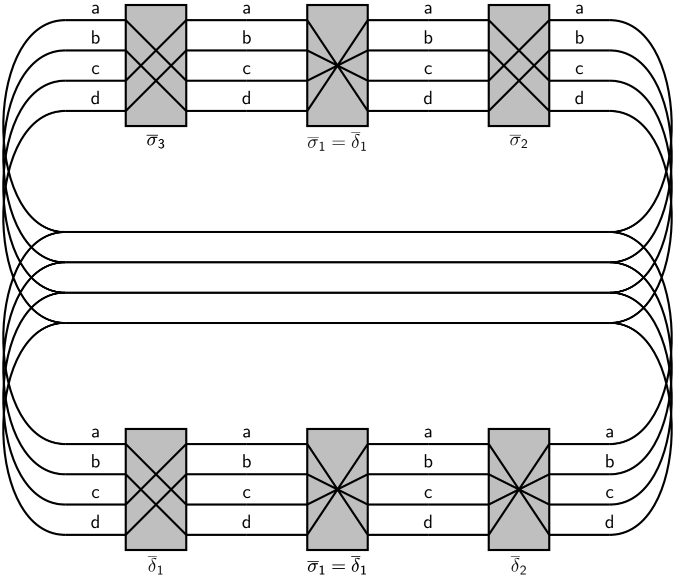
a
b
c
d
a
b
c
d
$\overline{\delta}_3$
A
C
a
b
c
d
a
b
c
d
a
b
c
d
$\overline{\delta}_1$
B
$\overline{\delta}_2$

Stops



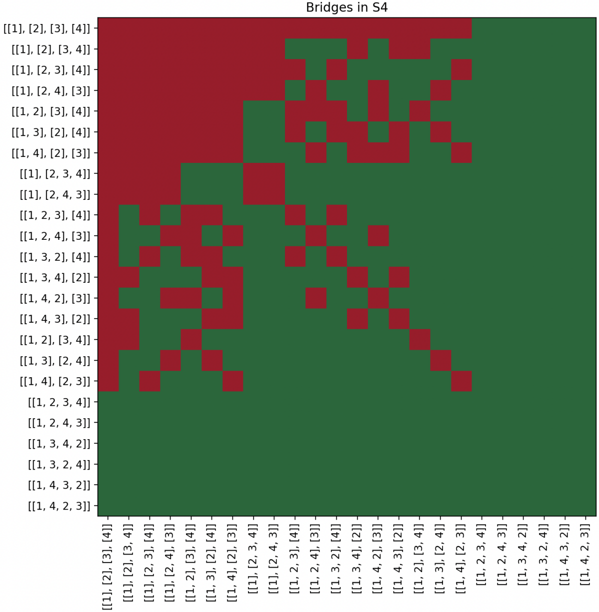
a
b
c
d
a
b
c
d
A
C
$\overline{\sigma}_3$
a
b
c
d
a
b
c
d
a
b
c
d
$\overline{\sigma}_1$
B
$\overline{\sigma}_2$

Stops



a
b
c
d
$\overline{\sigma}_3$
$\overline{\sigma}_1 = \overline{\delta}_1$
$\overline{\sigma}_2$
$\overline{\delta}_1$
$\overline{\sigma}_1 = \overline{\delta}_1$
$\overline{\delta}_2$

Stops



Bridges in S4
[[1], [2], [3], [4]]
[[1], [2], [3, 4]]
[[1], [2, 3], [4]]
[[1], [2, 4], [3]]
[[1, 2], [3], [4]]
[[1, 3], [2], [4]]
[[1, 4], [2], [3]]
[[1], [2, 3, 4]]
[[1], [2, 4, 3]]
[[1, 2, 3], [4]]
[[1, 2, 4], [3]]
[[1, 3, 2], [4]]
[[1, 3, 4], [2]]
[[1, 4, 2], [3]]
[[1, 4, 3], [2]]
[[1, 2], [3, 4]]
[[1, 3], [2, 4]]
[[1, 4], [2, 3]]
[[1, 2, 3, 4]]
[[1, 2, 4, 3]]
[[1, 3, 4, 2]]
[[1, 3, 2, 4]]
[[1, 4, 3, 2]]
[[1, 4, 2, 3]]