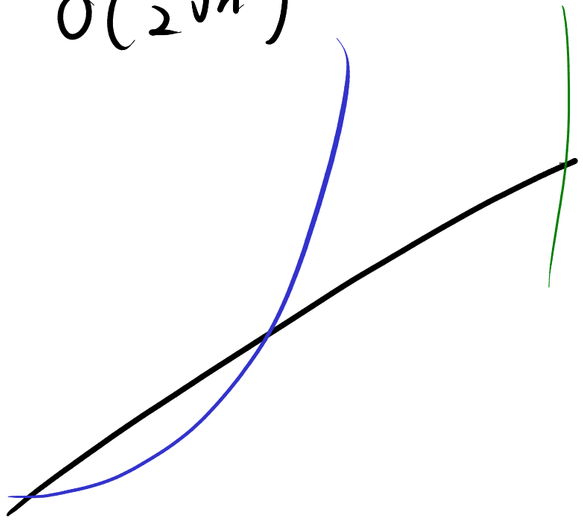
Complexity of algorithm

is # of steps it takes to execute

How does this scale w/ size?

$O(n)$, $O(n^c)$, $O(2^n)$, $O(2^{\sqrt{n}})$

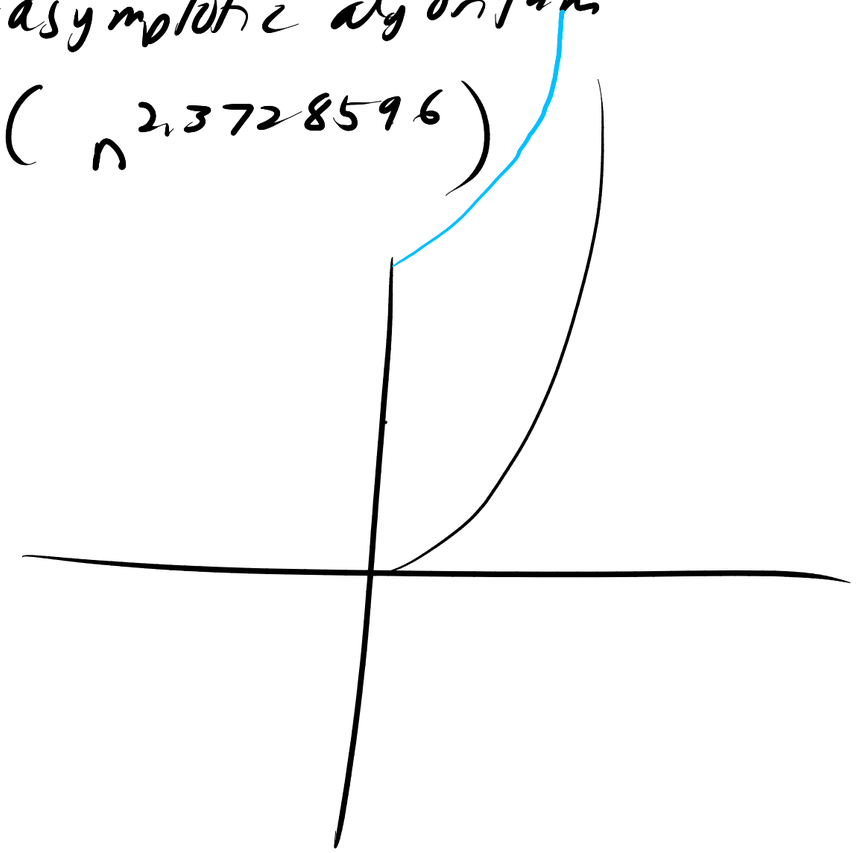


Matrix multiplication

Naive algorithm: $O(n^3)$

best asymptotic algorithm

$O(n^{2.3728596})$



Goal: One-way fn
easy to compute
hard to invert

One application: Stream cipher

$f: \{0,1\}^n \to (\mathbb{Z}/26\mathbb{Z})^{\mathbb{N}}$

sequences of integers mod 26

This gives me a fake one-time pad

fake b/c you can invert
can figure out my binary #
and thus my keystream.

But w/ one-way fn, very hard/
expensive to do this.

Big problem: do these exist?
We don't know.

$P \stackrel{?}{=} NP$

P = poly nomial time

NP = non-deterministic polynomial time

Random algorithm
we get perfectly lucky
then runs in poly time.

If f is in P
then inverting f is in NP.

NP ex! guess and check.

how to invert a 1-way fn

1) choose an input x at random
2) compute f(x)
3) is f(x) what you wanted?
   If yes, done.
   If no, start over.

i.e. $\sqrt[3]{30}$
equal to 3?
no, b/c $3^3 = 27 \neq 30$

$1 million if you answer this

NP-complete problems
if you can solve one, can solve any NP-problem

Coding Theory

predefined correspondences btwn messages and codes

We want to turn messages w/o #s.

1 approach

letters → $\mathbb{Z}/26\mathbb{Z}$

Problems:

- Clear what you're doing
- No spaces, or characters
- strings of #s.

000	␀	(nul)	016	►	(dle)	032	␣	048	0	064	@	080	P	096	‘	112	p
001	☺	(soh)	017	◄	(dc1)	033	!	049	1	065	A	081	Q	097	a	113	q
002	☻	(stx)	018	↕	(dc2)	034	"	050	2	066	B	082	R	098	b	114	r
003	♥	(etx)	019	‼	(dc3)	035	#	051	3	067	C	083	S	099	c	115	s
004	♦	(eot)	020	¶	(dc4)	036	$	052	4	068	D	084	T	100	d	116	t
005	♣	(enq)	021	§	(nak)	037	%	053	5	069	E	085	U	101	e	117	u
006	♠	(ack)	022	▬	(syn)	038	&	054	6	070	F	086	V	102	f	118	v
007	•	(bel)	023	↨	(etb)	039	'	055	7	071	G	087	W	103	g	119	w
008	◘	(bs)	024	↑	(can)	040	(	056	8	072	H	088	X	104	h	120	x
009		(tab)	025	↓	(em)	041	)	057	9	073	I	089	Y	105	i	121	y
010	◙	(lf)	026		(eof)	042	*	058	:	074	J	090	Z	106	j	122	z
011	♂	(vt)	027	←	(esc)	043	+	059	;	075	K	091	[	107	k	123	{
012		(np)	028	└	(fs)	044	’	060	<	076	L	092	\	108	l	124	|
013	♪	(cr)	029	↔	(gs)	045	-	061	=	077	M	093	]	109	m	125	}
014	♫	(so)	030	▲	(rs)	046	.	062	>	078	N	094	^	110	n	126	~
015	☼	(si)	031	▼	(us)	047	/	063	?	079	O	095	_	111	o	127	⌂

ASCII

$2^7 = 128$ symbols

P O K E 5 9 4 5

01010000 01001111 01001011 01000101 00100000 00110101 00111001 00110100 00110101 = 636279331279092264742596511083400111000 00101100 00110110 00110010

6 , 6 2

$01010000 = 2^4 + 2^6 = 80$

$2^7\ 2^6\ 2^5\ 2^4\ 2^3\ 2^2\ 2^1\ 2^0$

0

$01001111 = 1+2+4+8+64 = 79$

000	␀	(nul)	016	►	(dle)	032	␣	048	0	064	@	080	P	096	‘	112	p
001	☺	(soh)	017	◄	(dc1)	033	!	049	1	065	A	081	Q	097	a	113	q
002	☻	(stx)	018	↕	(dc2)	034	"	050	2	066	B	082	R	098	b	114	r
003	♥	(etx)	019	‼	(dc3)	035	#	051	3	067	C	083	S	099	c	115	s
004	♦	(eot)	020	¶	(dc4)	036	$	052	4	068	D	084	T	100	d	116	t
005	♣	(enq)	021	§	(nak)	037	%	053	5	069	E	085	U	101	e	117	u
006	♠	(ack)	022	▬	(syn)	038	&	054	6	070	F	086	V	102	f	118	v
007	•	(bel)	023	↨	(etb)	039	'	055	7	071	G	087	W	103	g	119	w
008	◘	(bs)	024	↑	(can)	040	(	056	8	072	H	088	X	104	h	120	x
009		(tab)	025	↓	(em)	041	)	057	9	073	I	089	Y	105	i	121	y
010	◙	(lf)	026		(eof)	042	*	058	:	074	J	090	Z	106	j	122	z
011	♂	(vt)	027	←	(esc)	043	+	059	;	075	K	091	[	107	k	123	{
012		(np)	028	└	(fs)	044	’	060	<	076	L	092	\	108	l	124	|
013	♪	(cr)	029	↔	(gs)	045	-	061	=	077	M	093	]	109	m	125	}
014	♫	(so)	030	▲	(rs)	046	.	062	>	078	N	094	^	110	n	126	~
015	☼	(si)	031	▼	(us)	047	/	063	?	079	O	095	_	111	o	127	⌂

$2^0 = 1$
$2^1 = 2$
$2^2 = 4$
$2^3 = 8$
$2^4 = 16$
$2^5 = 32$
$2^6 = 64$
$2^7 = 128$

encode Step

$83 = 64+16+2+1 \rightarrow 01010011$

$116 = 64+32+16+4 \rightarrow 01110100$

$111 = 64+32+8+4+2+1 \rightarrow 01101111$

$112 = 64+32+16 \rightarrow 01110000$

83 116 111 112

01010011 01110100 01101111 01110000

$2^{30}+2^{28}+2^{25}+2^{24}+2^{22}+2^{21}+2^{20}+2^{16}$
$+2^{14}+2^{13}+2^{11}+2^{10}+2^9+2^6+2^5+2^4$

$83+256^3+116\cdot256^2+111\cdot256+112\cdot256^0$

$= 1,400,136,632$

do a hill cipher mod 64

each # is an 8-byte block

Unicode: 2-byte blocks $(2^{16} = 65,536)$
3-byte blocks (2^{24})

☺ = 01F60A = 00000001 1111 0110 0000 1010

O big-O upper bound

$n = O(n^2)$

o little-o notation
Strong upper bound

$f = o(g)$ if $\lim \frac{f}{g} = 0$

$n = o(n^2)$

$n^2 \neq o(n^2)$

$\lim \frac{f}{g} = c \neq 0$

$100n^2 + 2^{100}n + 2^{2^{100}} = \Theta(n^2)$

this matters for crypto

Θ notation

$n^2 = \Theta(n^2)$

$n \neq \Theta(n^2)$