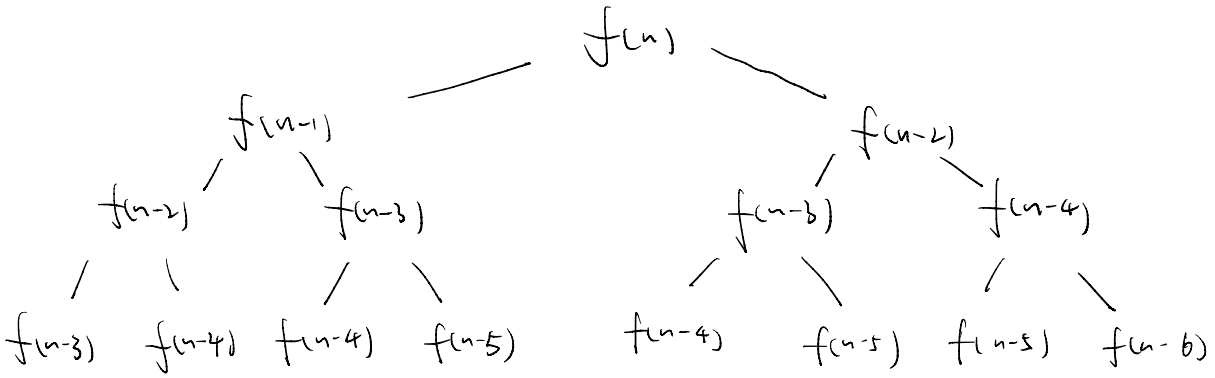
Lecture 8 March 29. 2013

1 1 2 3 5 8 13

$$f(n) = f(n-1) + f(n-2)$$

```
fib(n) {
    if ((n==0) || (n==1))
        return 1
    return (f(n-1) + f(n-2));
}
```




```
T[0] = 1        T[2-n] = -1
T[1] = 1
fib(n) {
    if T[n] != -1
        return T[n]
    T[n] = fib(n-1) + fib(n-2)
        return T[n]
```

Longest Common Subsequence. Leetcode 1143.

X

	-	A	C	G	G	C	A	T
-	0	0	0	0	0	0	0	0
A	0	1	1	1	1	1	1	1
T	0	1	1	1	1	1	1	2
C	0	1	2	2	2	2	2	2
G	0	1	2	3	3	3	3	3
G	0	1	2	3	4	4	4	4
A	0	1	2	3	4	4	5	5
T	0	1	2	3	4	4	5	6

Y

A	A
C	~~T~~
G	C
G	G
C	G
A	A → C
T	T

if $(y_i == x_j)$ $T[i-1, j-1]+1$

Else $\max[T[i-1, j], T[i, j-1]]$

ADDS + DELETE

Edit Distance: Levenshtein Edit Distance

.	—	A	C	G	G	C	A	T
-	0	1	2	3	4	5	6	7
A	1	0	1	2	3	4	5	6
T	2	1	1	2	3	4	5	5
C	3	2	1	2	3	3	4	5
G	4	3	2	1	2	3	4	5
G	5	4	3	2	1	2	3	4
A	6	5	4	3	2	2	2	3
T	7	6	5	4	3	3	3	2

Add

Delete

Substitute.

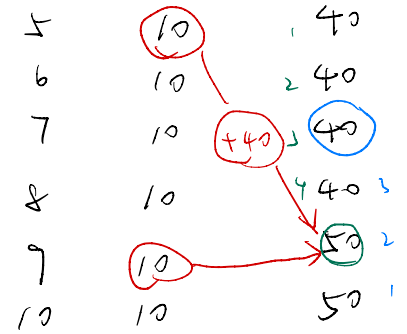
if $(y_i == x_j)$ $\min(T[i-1, j]+1, T[i, j-1]+1, T[i-1, j-1])$

Else $\min(T[i-1, j]+1, T[i, j-1]+1, T[i-1, j-1]+1)$

Knapsack

	I_1	I_2	I_3	I_4
Value	10	40	30	50
weight	5	4	3	3

weight I can take (pounds)	I_1	I_1+I_2	$I_1+I_2+I_3$
0	0	0	0
1	0	0	0
2	0	0	0
3	0	0	30
4	0	40	40
5	10	40	40
6	10	40	40
7	10	40	70
8	10	40	70
9	10	50	70
10	10	50	70



value

weight 3

Carry weight	Value for item 1 to 3	Value for items 1 to 4
0	$0	
1	$1	
2	$8	
3	$10	
4	$11	
5	$18	$18
6	$19	
7	$19	
8	$19	

Item #4
weighs $I_4.w = 4$ lbs
and has value $I_4.v = \$12$

If I can carry w = 5 lbs, should I take item #4?

If I don't take it, I can keep the $18 from items 1-3.

If I do take it, I have 1 lb left over from which I can get $1 value from 1-3

Formula: $\max(T[\text{prev}, W], T[\text{prevCol}, W-I4.w] + I4.v)$

For w = 5: $\max(18 \text{ or } 12+1)$ - Don't take I_4 for 5 lbs.

If w = 6 lb: $\max(19, 12+8) \Rightarrow$ Take I_4

	item #1	item #2	item #3	item #4
weight(lb)	2	3	1	4
value ($)	8	10	1	12

item #1, #2, #3, #4 only have one

carry weight	value for item 1	Value for items 1 and 2	Value for items 1 to 3	Value for items 1 to 4
0	0	0	0	0
1	0	0	1	1
2	8	8	8	8
3	8	10	10	10
4	8	10	11	12
5	8	18	18	18
6	8	18	19	20
7	8	18	19	22
8	8	18	19	23

Formula: Max(T[Prev, w], T[Prev Col, W-I4.w] + I4.w]

3 six-sided die {1, 2, 3, 4, 5, 6}

六面骰子

$6^3 = 216$ rolls possible.

sum \ die:	1	2
0	0	0
1	1	0
2	1	1
3	1	2
4	1	3
5	1	4
6	1	5
7	0	6
8	0	5
9	0	4
10	0	3
11	0	2
12	0	1
13	0	0
14	0	0
15	0	0

	prev Dice	Current Die
2	1	1
3	2 way × 1 way 1 way × 1 way 2	= 2 ways
4	3 way × 1 way 2 way × 1 way 2 1 way × 1 way 3	= 3 ways
5	4	1
	3	2
	2	3
	1	4
6	5	1
	4	2
	3	3
	2	4
	1	5
7	6	1
	5	2
	4	3
	3	4
	2	5
	1	6
8	6	2
	5	3
	4	4
	3	5
	2	6

⋮

sum \ die:	1	2	3	
0	0	0		
1	1	0		
2	1	1		
3	1	2		
4	1	3		
5	1	4		
6	1	5		
7	0	6		
8	0	5		
9	0	4		
10	0	3		
11	0	2		
12	0	1		
13	0	0		
14	0	0		
15	0	0		10

15

prev Dice		Current Die		
14	0	$\log 1$		
13	0	$\log 2$		
12	1	$\times$ $\log 3$		$= 1$
11	2	$\times$ $\log 4$		$= 2$
10	3	$\times$ $\log 5$		$= 3$
9	4	$\times$ $\log 6$		$= 4$

10

Greatest common divisor (GCD)

GCD (18, 12) = 6

GCD (641, 23) = $\left[\frac{641}{23}\right] = 27$ 641 % 23 = 20

GCD (23, 20)

GCD (20, 3)

GCD (3, 2)

GCD (2, 1)

GCD (1, 0)

	A	B	Q	R	α	β
-1					1	0
0	641	23	27	20	0	1
1	23	20	1	3	1	-27
2	20	3	6	2	-1	28
3	3	2	1	1	7	-195
4	2	1	2	0	-8	223
5	1	0	-	-	23	-641

$\beta_1 = \beta_{-1} - \beta_0 Q_0$

$= 0 - 1 \times 27$

$= -27$

$\alpha_1 = \alpha_{-1} - \alpha_0 Q_0$

$= 1 - 0 \cdot Q_0$

$= 1$

$\alpha_k = \alpha_{k-2} - \alpha_{k-1} Q_{k-1}$

$\beta_k = \beta_{k-2} - \beta_{k-1} Q_{k-1}$

$Q_k = \left[\frac{A_k}{B_k}\right]$

$R_k = A_k \% B_k$

$A_k = B_{k-1}$

$B_k = B_{k-1}$

$A_0 \cdot \alpha_k + B_0 \cdot \beta_k = B_k$

$-8 \times 641 + 23 \times 223 = 1$

$\frac{1}{23} \% 641 = 223$

$(23 \cdot 223) \% 641 = 1$

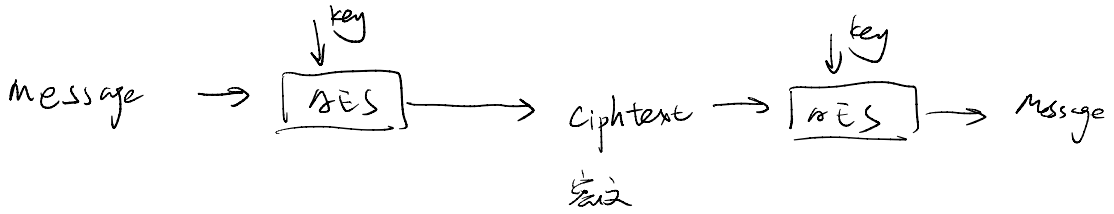
$\frac{1}{23} = 223$

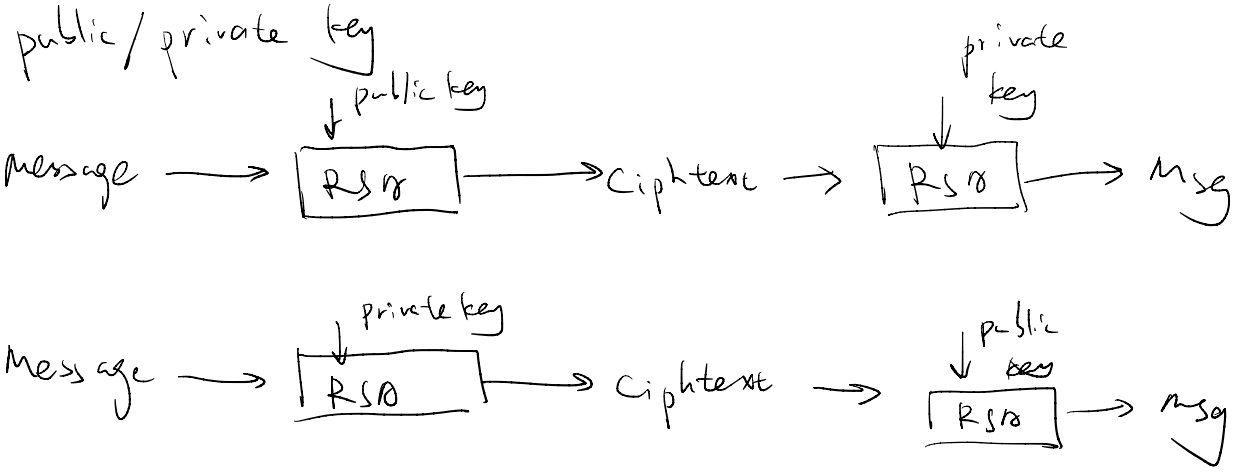
$(-8 \cdot 641) \% 641 + (23 \cdot 223) \% 641 = 1 \% 641$

$0 + (23 \cdot 223) \% 641 = 1$

	A	B	Q	R
0	18	12	1	6
1	12	6	2	0
2	6	0	—	—

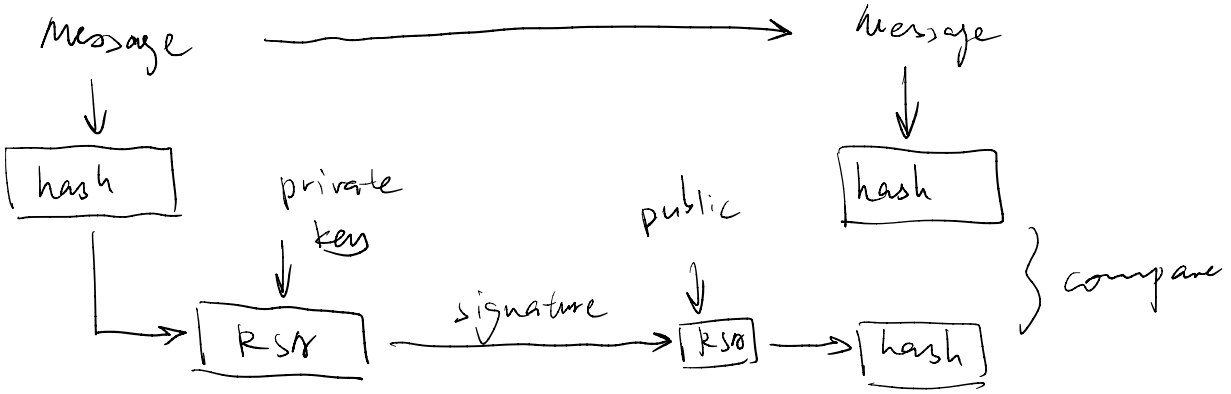
Symmetric key cryptography





public / private key is expensive. Usually is not happen.

Normally what happen is this:



Message is a number

8 MB Message is a number with 64 million bits.

You would break it into "chunks")

32000 - 2k bit chunks.

public key (e, n) private key (d, n)

encrypt $\text{message}^e \% n = \text{ciphertext}$

$\text{ciphertext}^d \% n = \text{message}$

sign $\left(\begin{matrix}\text{message} \\ \text{a} \\ \text{hash}\end{matrix}\right)^d \% n = \begin{matrix}\text{ciphertext} \\ \text{signature}\end{matrix}$

$\left(\begin{matrix}\text{ciphertext} \\ \text{signature}\end{matrix}\right)^e \% n = \left(\begin{matrix}\text{message} \\ \text{a} \\ \text{hash}\end{matrix}\right)$

$m^{ed} \% n = m$

$m^{\phi(n)} \% n = 1$

$m^{\phi(n)+1} \% n = m$ $\qquad ed \equiv \phi(n) + 1$

$ed \% \phi(n) = 1$

$2^1 \% 11 = 2$

$2^{11} \% 11 = 2$

$2^{21} \% 11 = 2$

$2^{31} \% 11 = 2$

$2^k \% 11 = 2$ if $k \% 10 = 1$

if $k \% \phi(11) = 1$

1. pick 2 prime numbers p, q

2. calculate $\boxed{n} = p \cdot q$

3. calculate $\phi(n) = (p-1)(q-1)$

4. pick $\boxed{e}$ such that $\gcd(\phi(n), e) = 1$ and $1 < e < \phi(n)$

5. calculate $\boxed{d} = \frac{1}{e} \% \phi(n)$

public key = $\{e, n\}$

private key = $\{d, n\}$

Forget p and q. if anybody figure out what p, q are, they can figure out what d is. So you got to lose p and e to protect d.

$(m^e)^d = m \qquad (m^d)^e \% n = m$

$\underline{c^{\boxed{d}} \% n = m}$

1. $p = 11 \qquad q = 17$

2. $n = 187 \qquad (11 \cdot 17)$

3. $\phi(n) = (p-1)(q-1) = 10 \cdot 16 = 160$

4. pick $e = 7$ public key $(7, 187)$

5. computed $d = \frac{1}{7} \% 160 = 23$ private key $(23, 187)$

r	b	q	R	α	β
				1	0
160	7	22	6	0	1
7	6	1	1	1	-22
6	1	6	0	-1	23

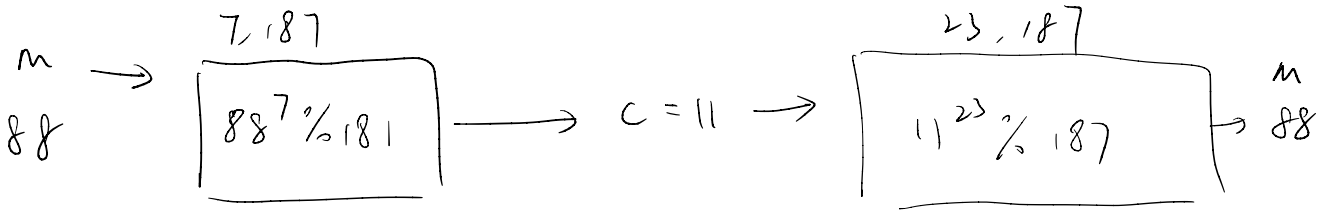
$\frac{1}{7} \% 160 = 23$

message = 88

$88^7 \% 187 = 11$

cipher + ext = 11

$11^{23} \% 187 = 88$



Test hit.

table. row may be just few steps. not too complex.

Longest Increasing Subsequence

5 1 8 9 3 2

5
1
8
9
3
2

1 8 9

5 2 8 1 9 3

5 5
2 2
8 2 8
1 1 8
9 1 8 9
3 1 3 9

2 8 9