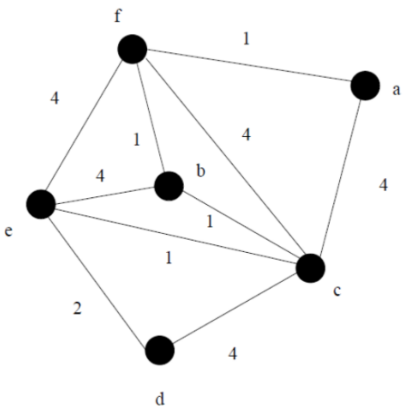
1, Degree of a vertex (看 vertex 有几个 edges)
2, Simple path: path such that all its vertices and edges are distinct.
3, A walk in a graph is a sequence of alternating vertices and edges, starting at a vertex and ending at a vertex.
A trail: a walk with no repeated edge.
A circuit is a walk with the same start and end vertex.
A cycle is a circuit where each vertex in the circuit is distinct (except for first and last vertex).

4, $\sum_v \deg(v) = 2m$

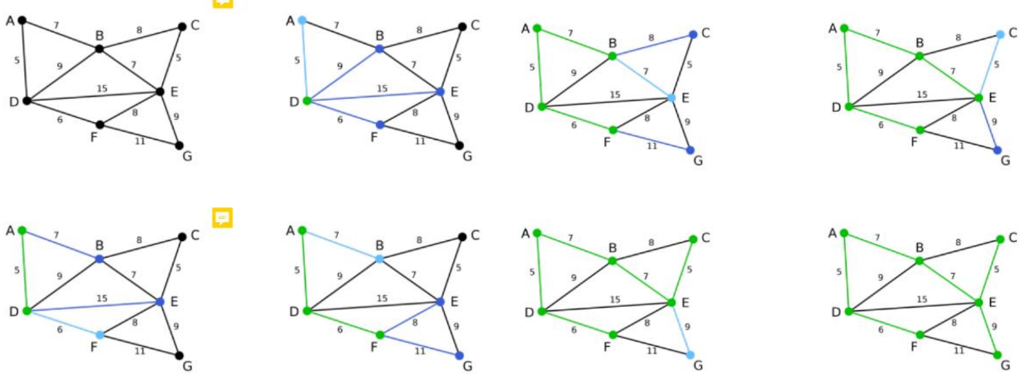
5,

Dijkstra's algorithm

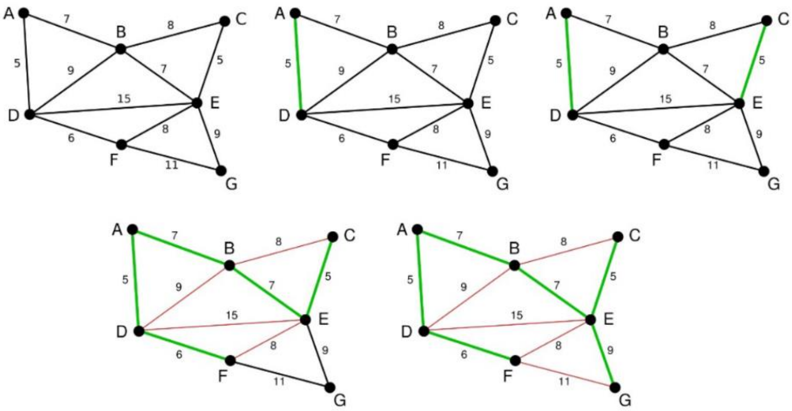


D [	a	b	c	d	e	f]	C
	∞	∞	∞	0	∞	∞	ϕ
	∞	∞	4	**0**	2	∞	$\{d\}$
	∞	6	3	**0**	**2**	6	$\{d, e\}$
	7	4	**3**	**0**	**2**	6	$\{c, d, e\}$
	7	**4**	**3**	**0**	**2**	5	$\{b, c, d, e\}$
	6	**4**	**3**	**0**	**2**	**5**	$\{b, c, d, e, f\}$
	6	**4**	**3**	**0**	**2**	**5**	$\{a, b, c, d, e, f\}$

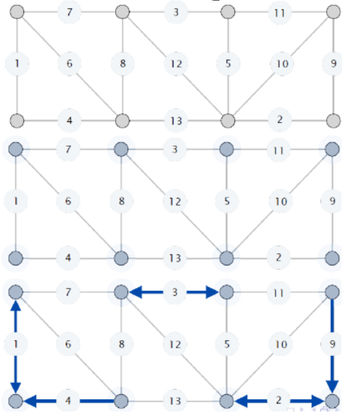
6, Prim's Algorithm



7, Kruskal's Algorithm for MST

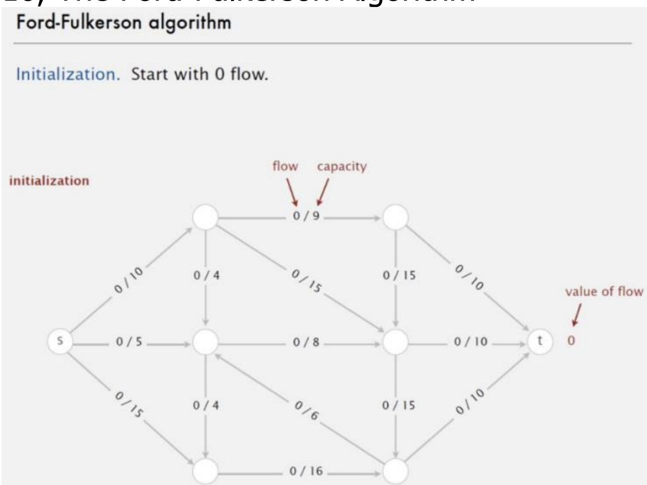


8, Borůvka's algorithm

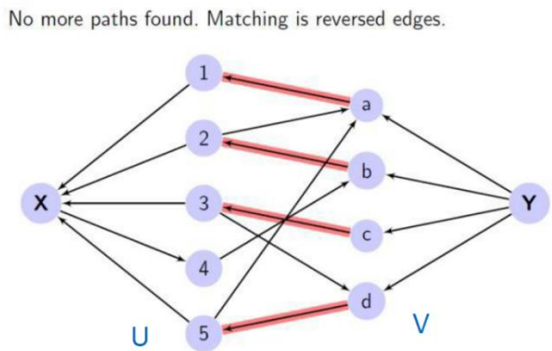


下面的图中，为方便计算将节点从左到右从上到下命名为abcdefgh。
第一步，将八个节点看作八颗树。
第二步，寻找距离每棵树距离最近的子树：
距离a最近的是e，距离b最近的是c，距离c最近的是b，距离的d最近的是h，距离e最近的是a，距离f最近的是a，距离g最近的是h，距离h最近的是g。
第三步，e和af相连，b和c相连，h和dg相连，这样就重新形成了三棵树。
第四步，将上述三棵树分别命名为1，2，3，距离1树最近的树是2树距离为7，距离2树最近的树是3树距离为5，距离3树最近的是2树距离为5。
最后一步，将三棵树按最短距离相连，得到MST。

9, Flow f (χ) across a cut χ: total flow of forward edges minus total flow of backward edges
Capacity c(χ) of a cut χ: total capacity of forward edges
10, The Ford-Fulkerson Algorithm



11, Maximum Bipartite Matching



12, 整除： n | a, 不能整除： $n \nmid a$

Property 1: if a|1, then a = ±1.

Property 2: if b|a and a|b, then a = ±b.

Property 3: if b|a and c|b, then c|a.

Property 4: if a|b and a|c, then a| (m × b + n × c), where m and n are arbitraryintegers

13,

14,

Note **Euclidean Algorithm**

Fact: gcd (a, 0) = a

Lemma: Let a, b, q, and r be integer such that $a = bq + r$ and $b \neq 0$. Then gcd(a, b) = gcd(b, r).

15, When gcd (a, b) = 1, we say that a and b are relatively prim.

16,

Basic Euclidean algorithms

```
def gcd(a,b)
    assert a>=b and b>=0 and a+b>0
   return gcd(b, a%b) if b>0 else a
```

Extended Euclidean Algorithm

$$s \times a + t \times b = \gcd(a, b)$$

$r_1 \leftarrow a$; $r_2 \leftarrow b$;
$s_1 \leftarrow 1$; $s_2 \leftarrow 0$; (Initialization)
$t_1 \leftarrow 0$; $t_2 \leftarrow 1$;

while ($r_2 > 0$)

{

$q \leftarrow r_1 / r_2$;

$r \leftarrow r_1 - q \times r_2$;
$r_1 \leftarrow r_2$; $r_2 \leftarrow r$; (Updating r's)

$s \leftarrow s_1 - q \times s_2$;
$s_1 \leftarrow s_2$; $s_2 \leftarrow s$; (Updating s's)

$t \leftarrow t_1 - q \times t_2$;
$t_1 \leftarrow t_2$; $t_2 \leftarrow t$; (Updating t's)

}

17, $\gcd(a, b) \leftarrow r_1$; $s \leftarrow s_1$; $t \leftarrow t_1$

18, Zn, Z6 = {0, 1, 2, 3, 4, 5}

First Property: $(a + b) \bmod n = [(a \bmod n) + (b \bmod n)] \bmod n$

Second Property: $(a - b) \bmod n = [(a \bmod n) - (b \bmod n)] \bmod n$

19, **Third Property:** $(a \times b) \bmod n = [(a \bmod n) \times (b \bmod n)] \bmod n$

$$a = a_n \times 10^n + \cdots + a_1 \times 10^1 + a_0 \times 10^0$$

10^n mod x = (10 mode x)^n mod x

20, Inverse:

Additive Inverse:

In Zn, two numbers a and b are additive inverses of each other if: a + b ≡ 0 (mod n)

Multiplicative Inverse:
In Zn, two numbers a and b are the multiplicative inverse of each other if: a x b ≡ 1 (mod n), a = b^{-1}
no multiplicative inverse if gcd (10, 8) = 2 ≠ 1.

21,

$r_1 \leftarrow n; \quad r_2 \leftarrow b;$
$t_1 \leftarrow 0; \quad t_2 \leftarrow 1;$

while ($r_2 > 0$)
{
$q \leftarrow r_1 / r_2;$

$r \leftarrow r_1 - q \times r_2;$
$r_1 \leftarrow r_2; \quad r_2 \leftarrow r;$

$t \leftarrow t_1 - q \times t_2;$
$t_1 \leftarrow t_2; \quad t_2 \leftarrow t;$
}
if ($r_1 = 1$) then $b^{-1} \leftarrow t_1$

22, Zn*: Zn 中所有和 n 互质的数 (gcd(a,n)=1)
23, single-Variable Linear Equations

Equations of the form $ax \equiv b \pmod n$ might have no solution or a limited number of solutions.

$\boxed{\text{Assume that the gcd } (a, n) = d.}$

If $d \nmid b$, there is no solution.

If $d | b$, there are d solutions.

Example

Solve the equation $10x \equiv 2 \pmod{15}$.

Solution

First we find the gcd (10, 15) = 5. Since 5 does not divide 2, we have no solution.

Example

Solve the equation $14x \equiv 12 \pmod{18}$.

Solution

Division in Z_n^* is defined by the equation $a/b \equiv ab^{-1} \pmod n$.

$14x \equiv 12 \pmod{18} \rightarrow \quad 7x \equiv 6 \pmod 9 \quad \rightarrow \quad x \equiv 6\ (7^{-1}) \pmod 9$
$x_0 = (6 \times 7^{-1}) \bmod 9 = (6 \times 4) \pmod 9 = 6$
$x_1 = x_0 + 1 \times (18/2) = 15$

24,

$3^{94} \pmod{17}$

Any number can be represented as the sum of distinct powers of two.

94=64+16+8+4+2

mod 17

$3^2 \equiv 9$
$3^4 \equiv 81 \equiv 13 \equiv -4$
$3^8 \equiv (3^4)^2 \equiv 16 \equiv -1$
$3^{16} \equiv (3^8)^2 \equiv (-1)^2 \equiv 1$
$3^{64} \equiv (3^{16})^4 \equiv (1)^4 \equiv 1$

Use the smallest numbers whether they are positive or negative

Theorem [Fermat's Little Theorem] : Let p be prime, and let x be an integer such that $x \bmod p \neq 0$. Then

$$x^{p-1} \equiv 1 \mod p$$

Corollary

Let p be a prime. For each nonzero residue x of Z_p, the multiplicative inverse of x is $x^{p-2} \bmod p$

Proof

$x(x^{p-2} \bmod p) \bmod p = xx^{p-2} \bmod p = x^{p-1} \bmod p = 1$

25,

$$x^{-1} \equiv x^{p-2} \ (\mathrm{mod}\, p)$$

26, Euler's function
ϕ(n) 是 Zn* 的长度

Let the prime factorisation of n is given by
$n=p_1^{e1}*\ldots*p_n^{en}$, then $\varphi(n) = n*(1-1/p_1)*\ldots*(1-1/p_n)$.

Theorem [Euler's Theorem] : Let n be a positive integer, and let x be an integer such that $\gcd(x, n) = 1$. Then

$$x^{\phi(n)} \equiv 1 \mod n$$

27, plaintext: 明文, ciphertext: 加密文
28, RSA encryption scheme:
Let $n = p \cdot q$ and define $\varphi(n) = (p - 1)(q - 1)$.

We then choose two numbers e and d such that

1. e and $\varphi(n)$ are relatively prime, i.e. $\gcd(e, \varphi(n)) = 1$
2. $ed \equiv 1 \pmod{\varphi(n)}$ (by Extended Euclidean algorithm)

◆Setup:
- $n = pq$, with p and q primes
- e relatively prime to $\phi(n) = (p-1)(q-1)$
- d inverse of e in $Z_{\phi(n)}$

◆Keys:
- Public key: $K_E = (n, e)$
- Private key: $K_D = d$

◆Encryption:
- Plaintext M in Z_n
- $C = M^e \bmod n$

◆Decryption:
- $M = C^d \bmod n$

◆Example
- Setup:
  - $p = 7,\ q = 17$
  - $n = 7 \cdot 17 = 119$
  - $\phi(n) = 6 \cdot 16 = 96$
  - $e = 5$
  - $d = 77$
- Keys:
  - public key: $(119, 5)$
  - private key: 77
- Encryption:
  - $M = 19$
  - $C = 19^5 \bmod 119 = 66$
- Decryption:
  - $M = 66^{77} \bmod 119 = 19$

29, Digital signatures:
RSA cryptosystem supports *digital signatures*. Suppose that Bob sends a message M to Alice and that Alice wants to *verify* that it was Bob who sent it. Bob can create a *signature* using the decryption function applied to M:

$$S \leftarrow M^d \bmod n.$$

Alice verifies the digital signature using the encryption function, that is by checking that

$$M \equiv S^e \pmod{n}.$$

30, NPC 问题：存在这样一个 NP 问题，所有的 NP 问题都可以约化成它。换句话说，只要解决了这个问题，那么所有的NP问题都解决了。
其定义要满足2个条件:
首先，它得是一个NP问题;

然后，所有的NP问题都可以约化到它。

要证明npc问题的思路就是： 先证明它至少是一个NP问题，再证明其中一个已知的NP 问题能约化到它。

31, 如果 L 可以在多项式时间内解出来，并且 L 里的 s 可以通过一个函数 f(s) 转变到 M 里，那么 L 就可以被约化到 M

$$L \xrightarrow{poly} M$$

32, NP-Hard问题是这样一种问题，它满足NPC问题定义的第二条但不一定要满足第一条（就是说，NP-Hard问题要比 NPC问题的范围广，NP-Hard问题没有限定属于NP），即所有的NP问题都能约化到它，但是它不一定是一个NP问题。

33, Conjunctive Normal Form:

$$(\overline{x_1} \vee \overline{x_2} \vee x_4 \vee \overline{x_6}) \wedge (\overline{x_2} \vee x_4 \vee \overline{x_5} \vee x_3)$$

3-SAT is CNF-SAT in which each clause has exactly three literals.

34, Approximation Ratios:

T is a k-approximation to the optimal solution OPT if c(T)/c(OPT) ≤ k (assuming a min. problem)

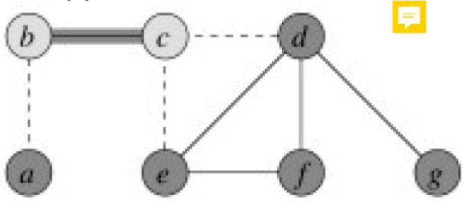
T is a k-approximation to the optimal solution OPT if c(OPT)/c(T) ≤k (assuming a max. problem)

The value of k is never less than 1.

34, Polynomial-Time Approximation Schemes: PTAS 的运行时间必须是 n 的多项式，但是它可以是 ε 的指数。

fully polynomial-time approximation scheme: fully PTAS 的运行时间不但要是是 n 的多项式，也要是 1 / Є 的多项式。

35, Approx-Vertex-Cover:



随便找个边，把两个顶点都加进结果集，将与这两个顶点相连的边都从候选边中移除。

36, Triangle Inequality TSP:

The algorithm finds a minimum spanning tree, and then apply pre-order traversal