

A binary tree is *proper* if each internal node has *exactly two children*.

Two nodes that are children of the same parent are called siblings.

The depth of a node, v, is number of ancestors of v, excluding v itself.

The height of a tree is equal to the maximum depth of an external node

in it.

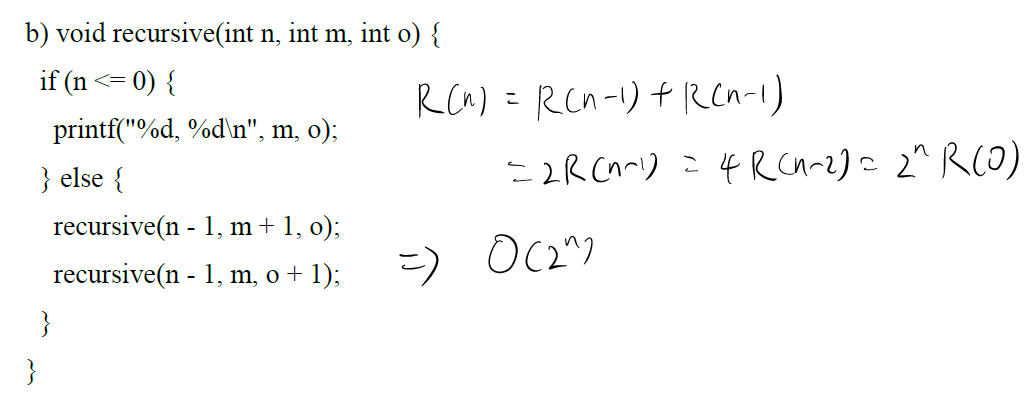
preorder, postorder, inorder:

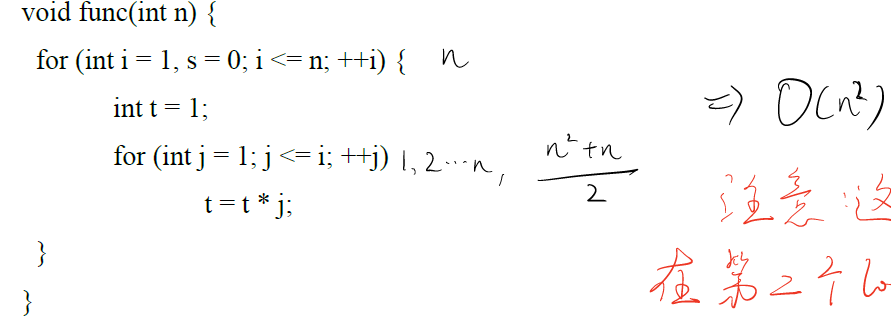
图示

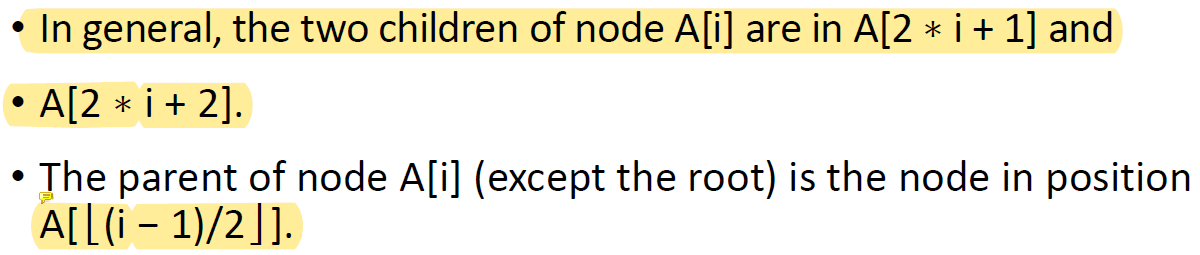
描述已自动生成 图示

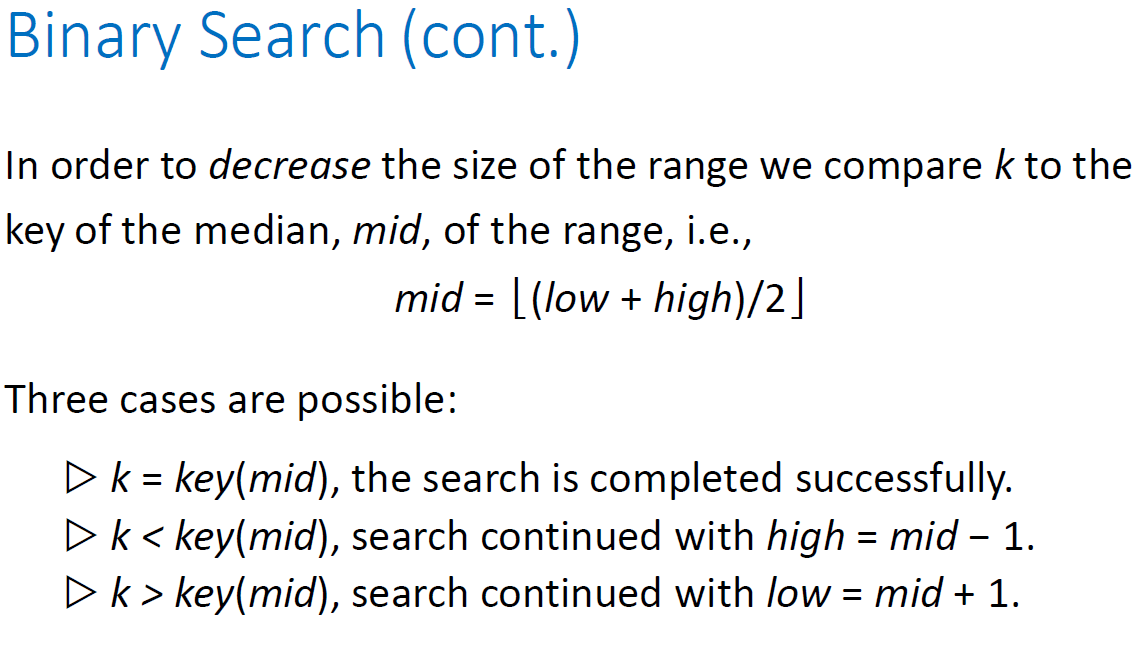
描述已自动生成 钟表的特写

描述已自动生成

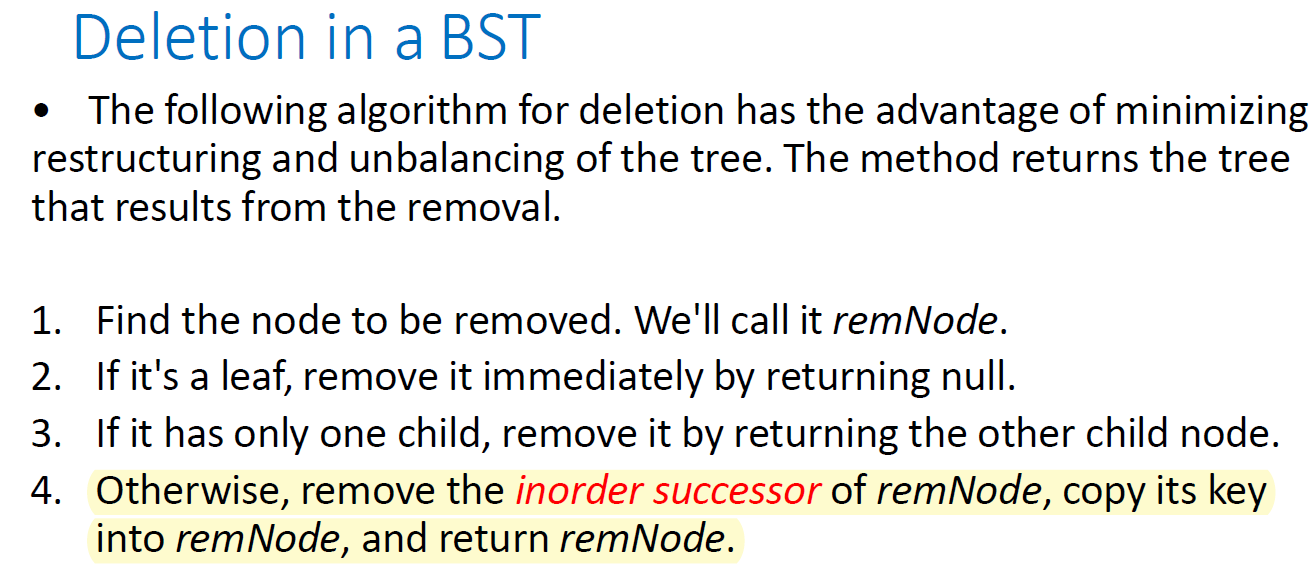












Deletion in BST (3 cases: leaf, one child, otherwise)

图示

描述已自动生成 图示

描述已自动生成

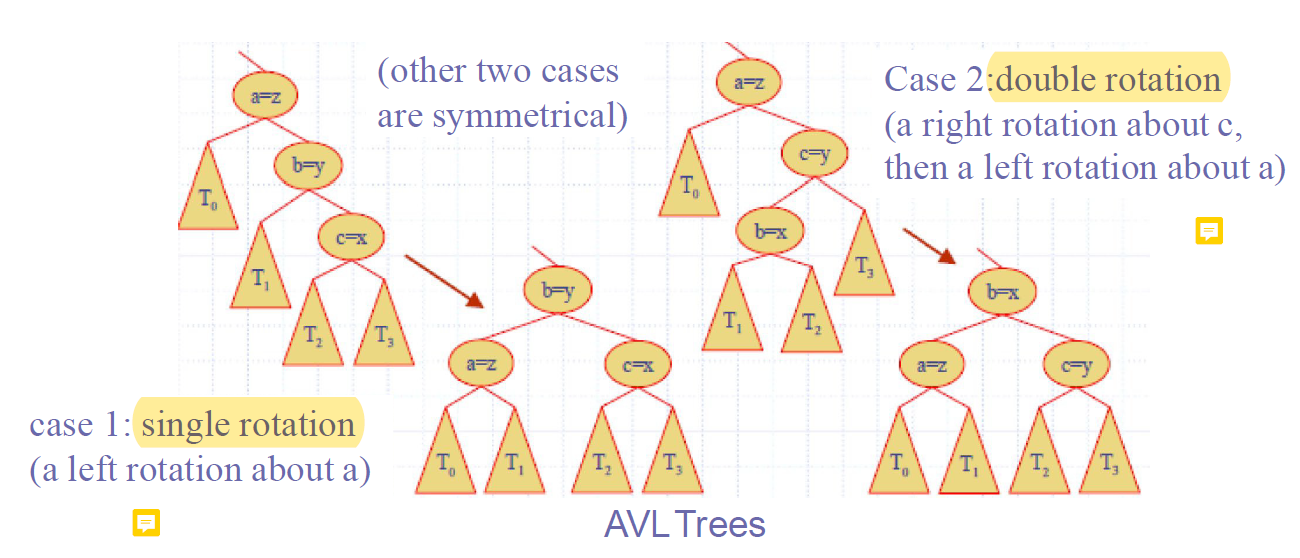
Otherwise, use inorder successor of removeElement to replace:

图示, 示意图

描述已自动生成

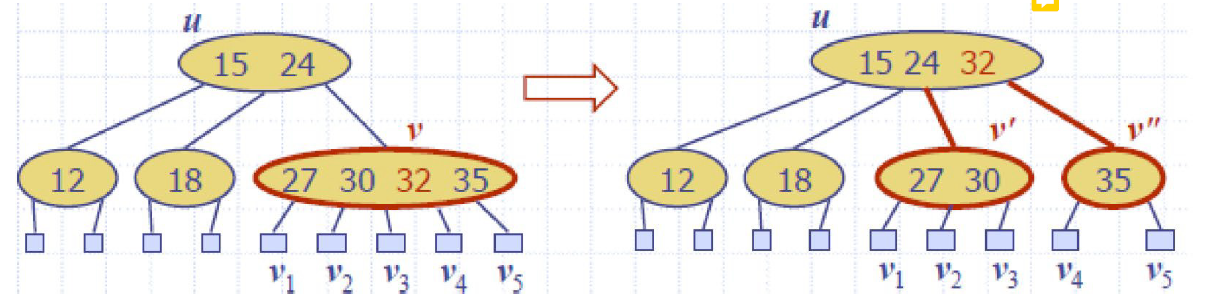
AVL tree：

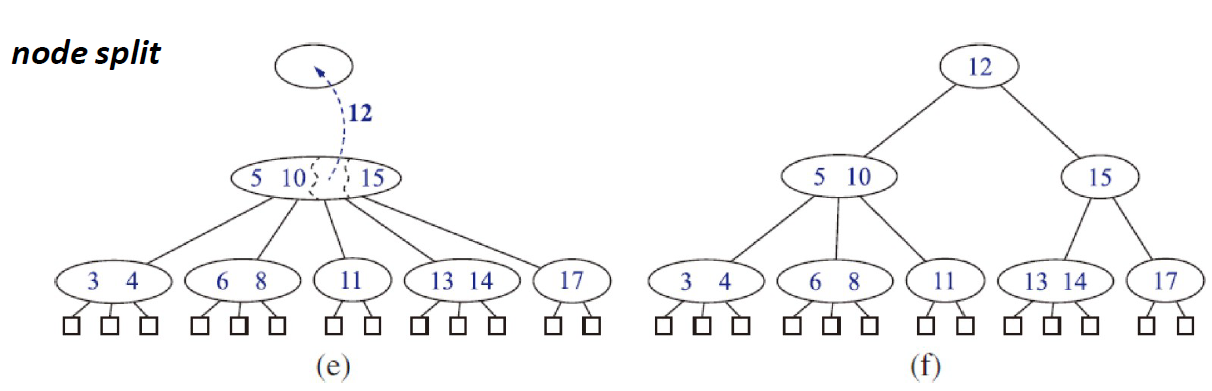
find the first node x such that its grandparent z is unbalanced node



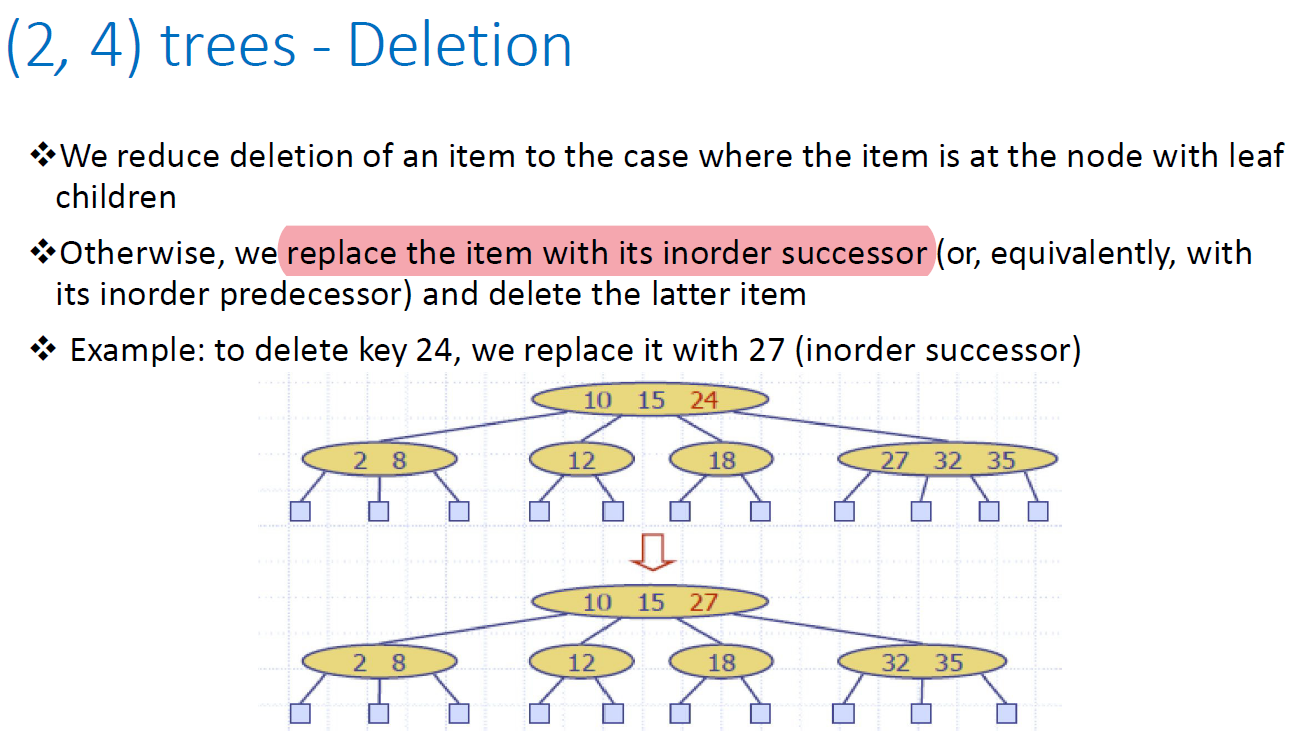
(2, 4)trees: all the external nodes have the same depth

Insertion：

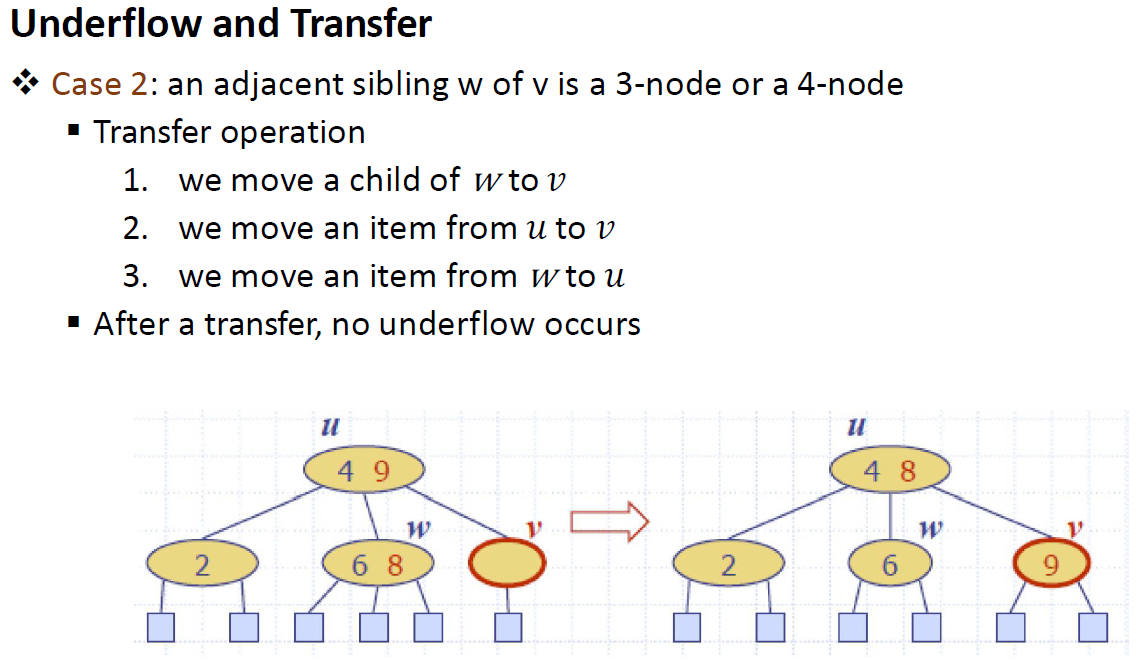


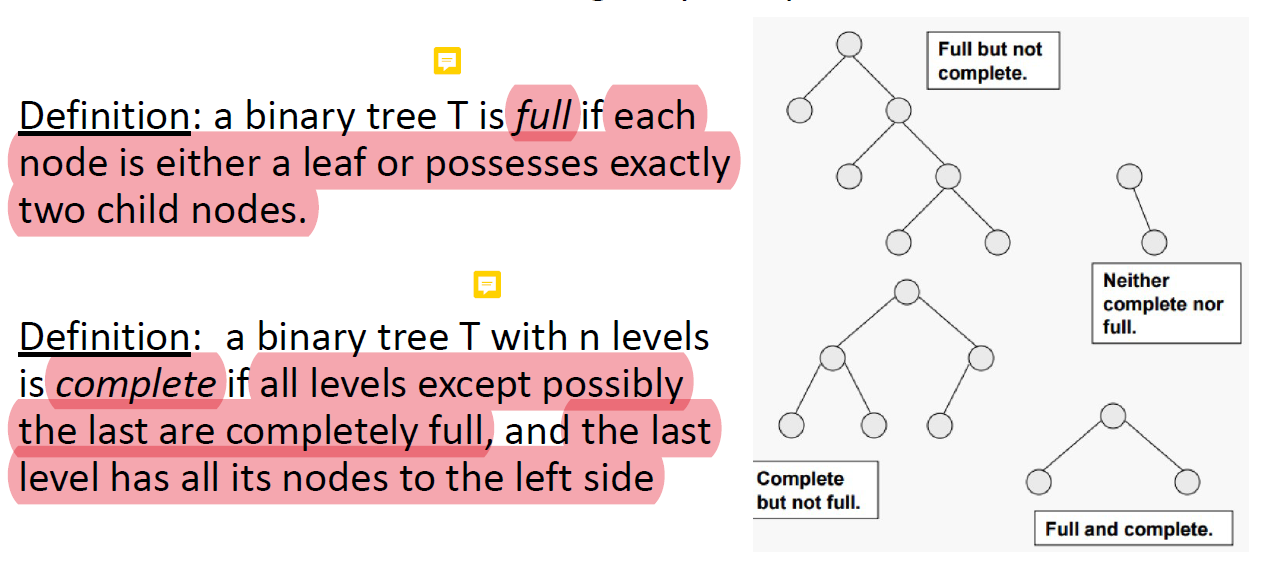


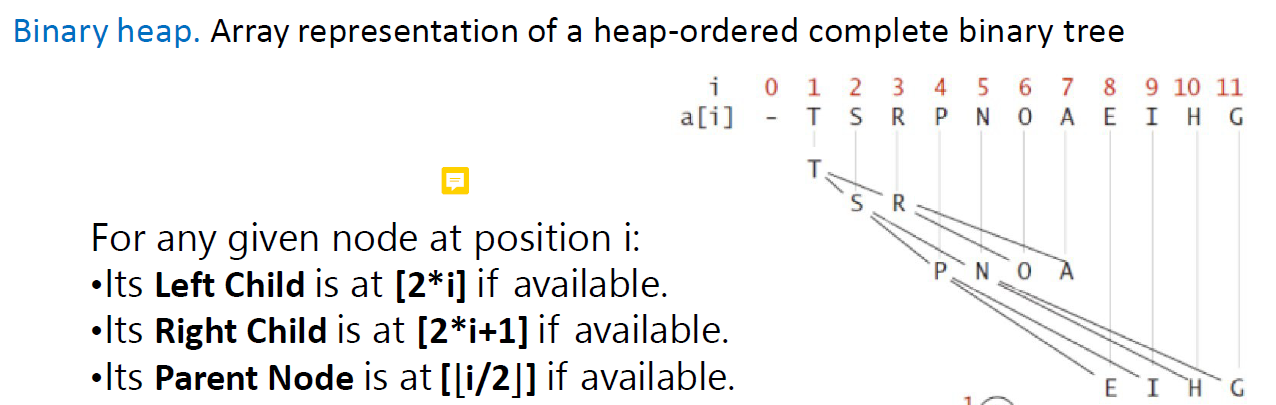
Deletion:











insertion in heap (O(logn))

图示

描述已自动生成 图示

描述已自动生成 图示

描述已自动生成

deletion in heap (removeMin for min-heap: replace root key with **last node**) (O(logn))

图示

描述已自动生成 图示

描述已自动生成 图示

描述已自动生成

MergeSort: O(nlogn), n + 2 \* n/2 + 4 \* n/4 + ... = n \* logn (height is logn) = nlogn

QuickSort: O(nlogn), worst-case: O(n2)

The Master method

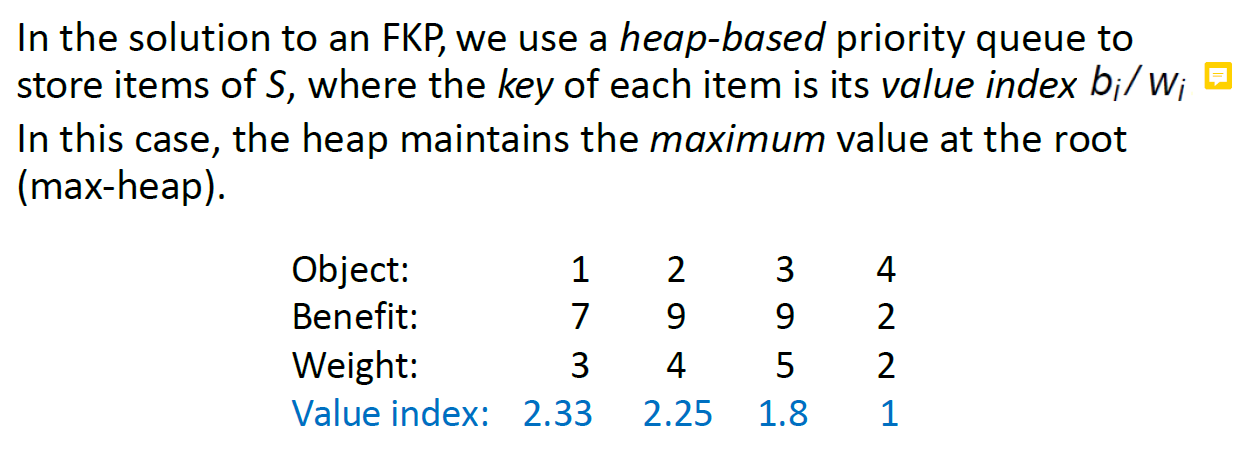
图片包含 文本

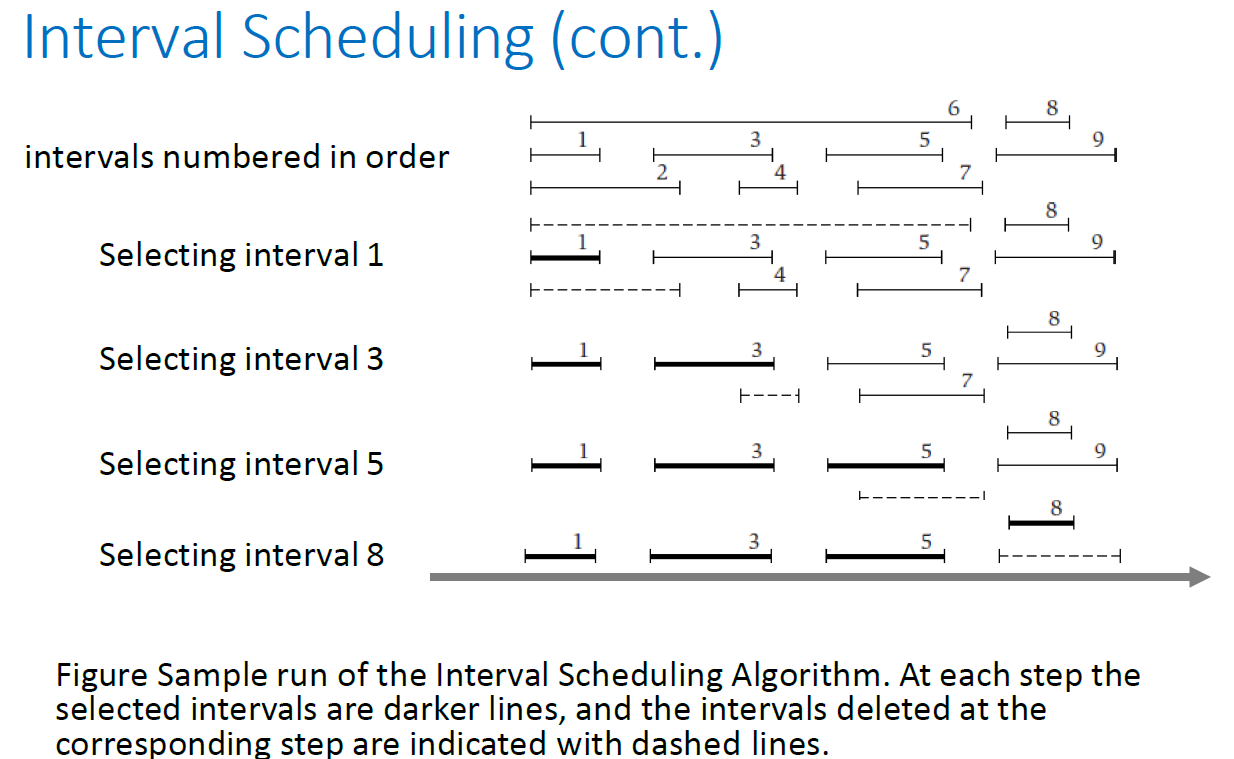
描述已自动生成 文本, 信件

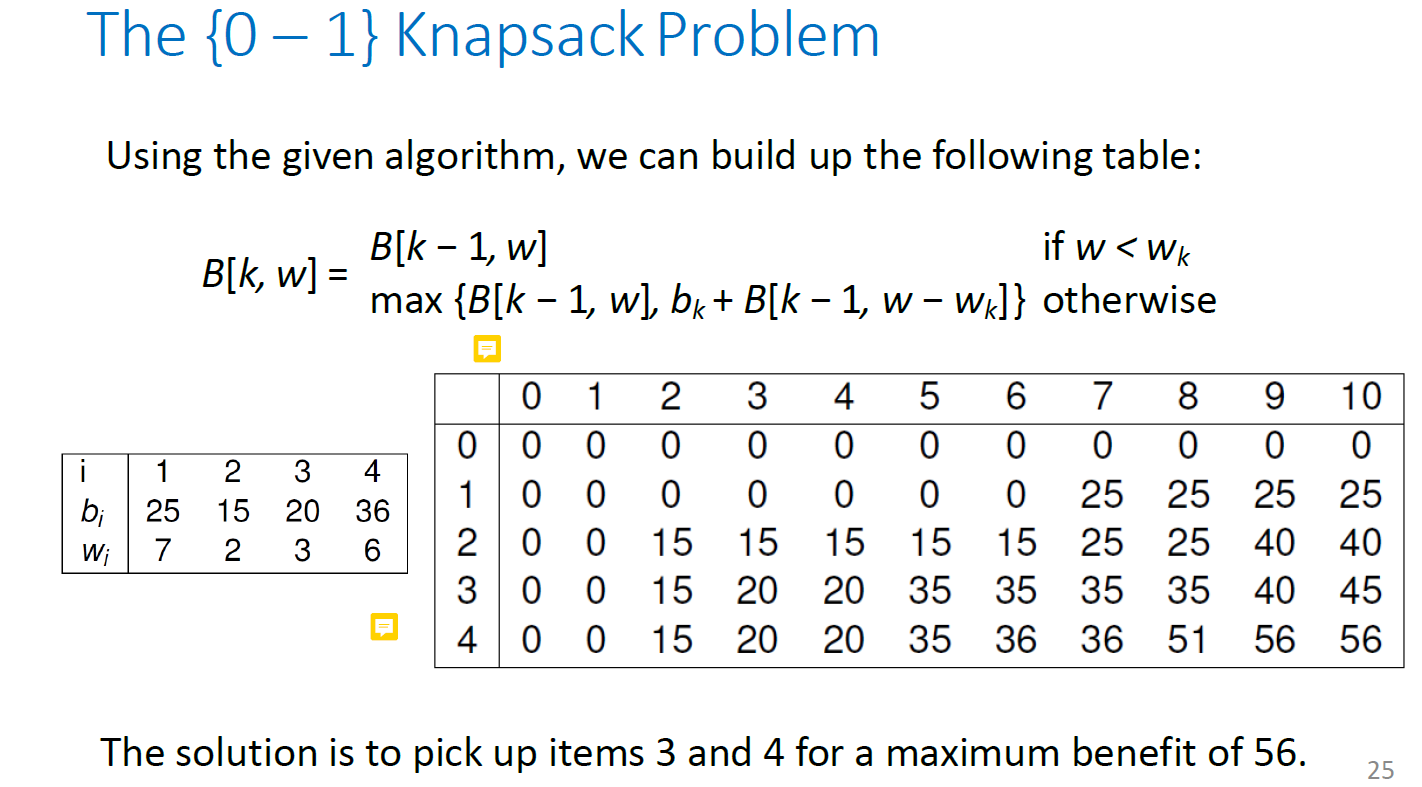
描述已自动生成

文本

描述已自动生成







等比等差求和: 钟表的特写

中度可信度描述已自动生成 文本

中度可信度描述已自动生成

求和:图表

中度可信度描述已自动生成手机屏幕截图

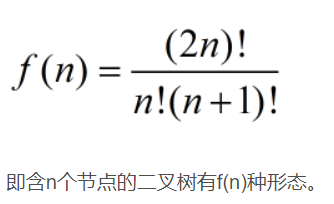
中度可信度描述已自动生成

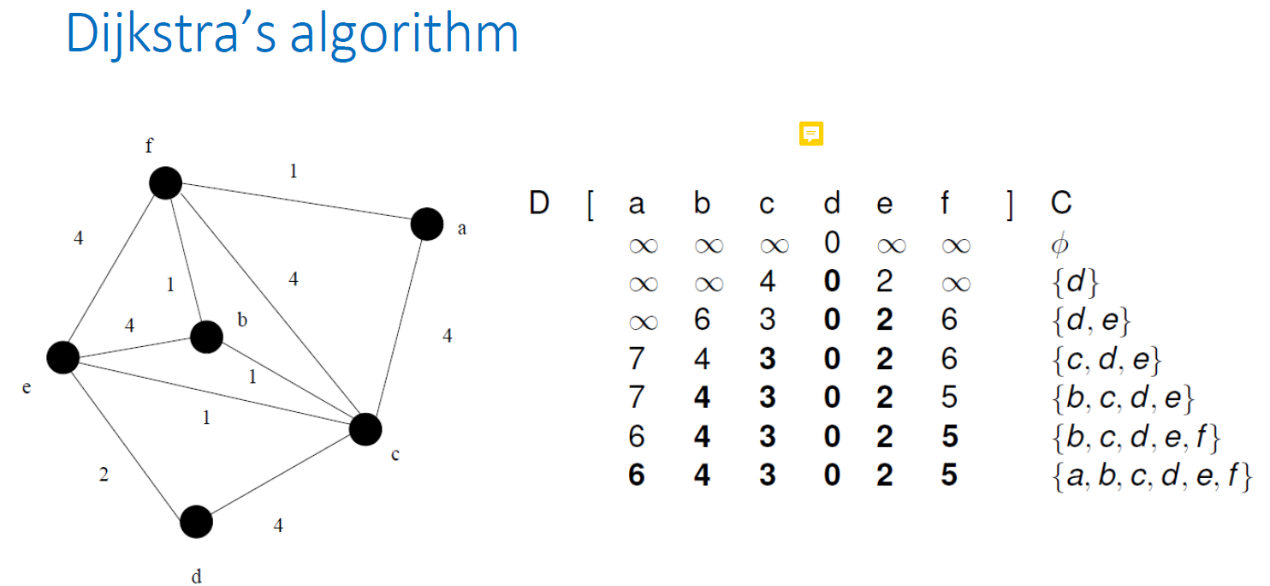
对数运算: 文本, 信件

描述已自动生成 文本, 信件

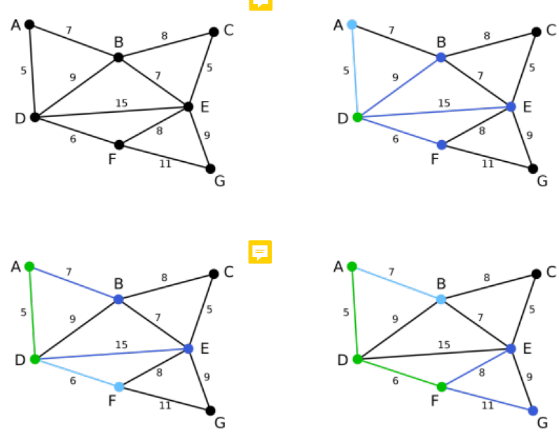
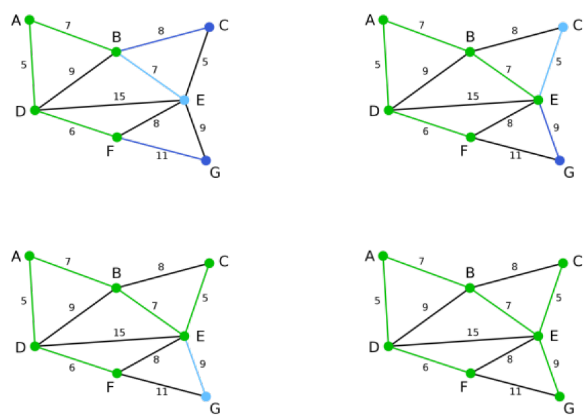
描述已自动生成 文本

中度可信度描述已自动生成

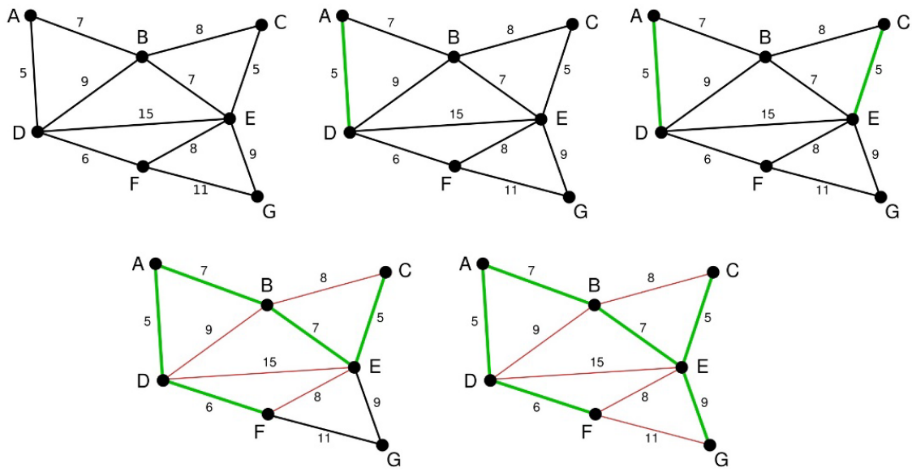




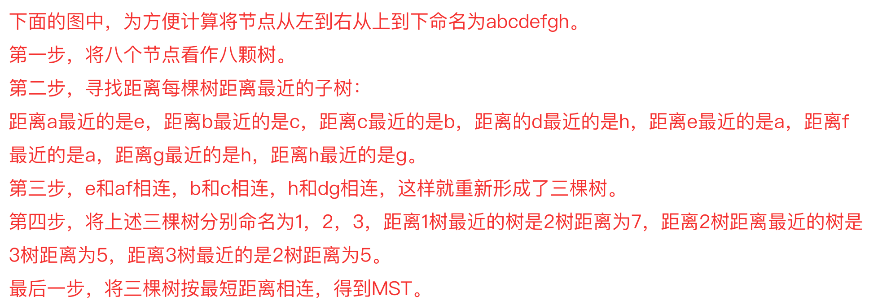
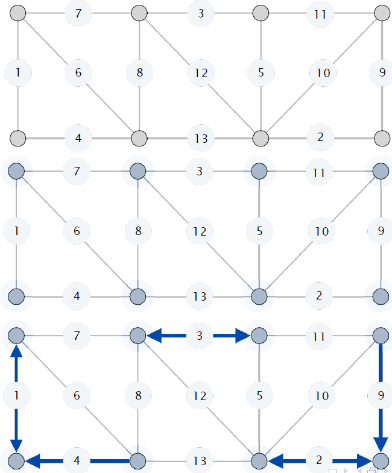
Prim’s Algorithm

Kruskal’s Algorithm for MST



Borůvka's algorithm



Flow f (χ) across a cut χ: total flow of forward edges minus total flow of backward edges

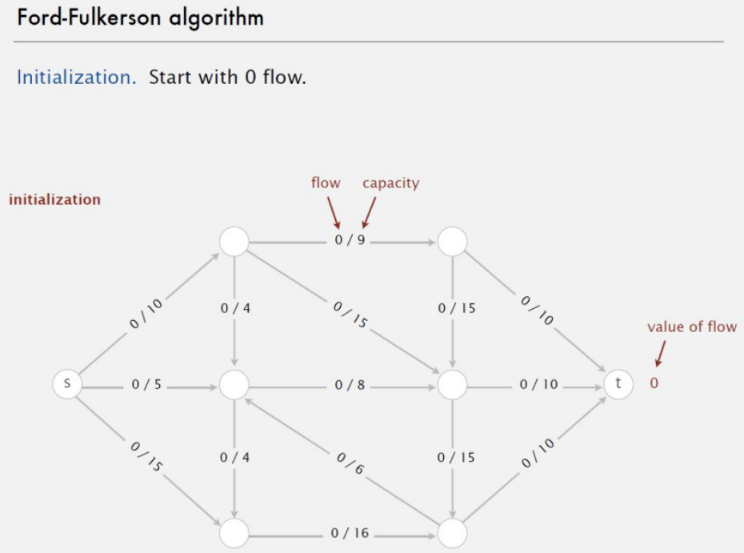
Capacity c(χ) of a cut χ: total capacity of forward edges

Min-Cut 的两个条件：

Forward edge: f(e) = c(e)

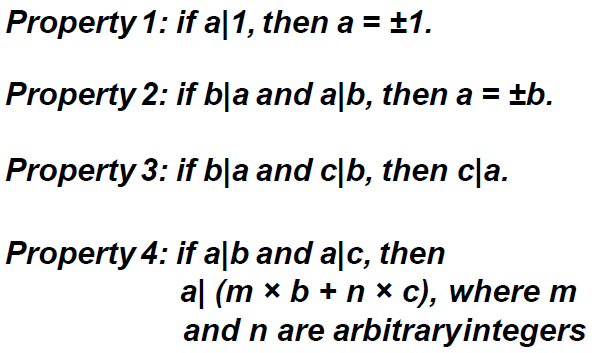
Backward edge: f(e) = 0

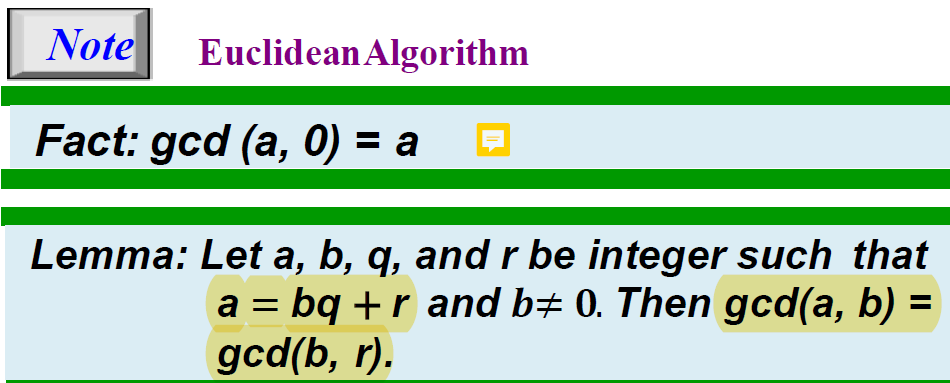
The Ford-Fulkerson Algorithm



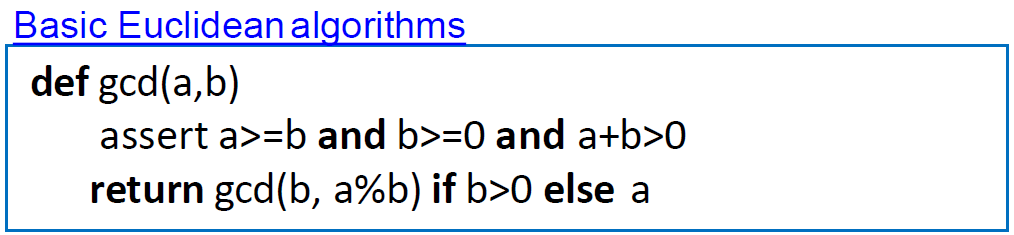
A Bipartite matching is perfect if no vertex is exposed.

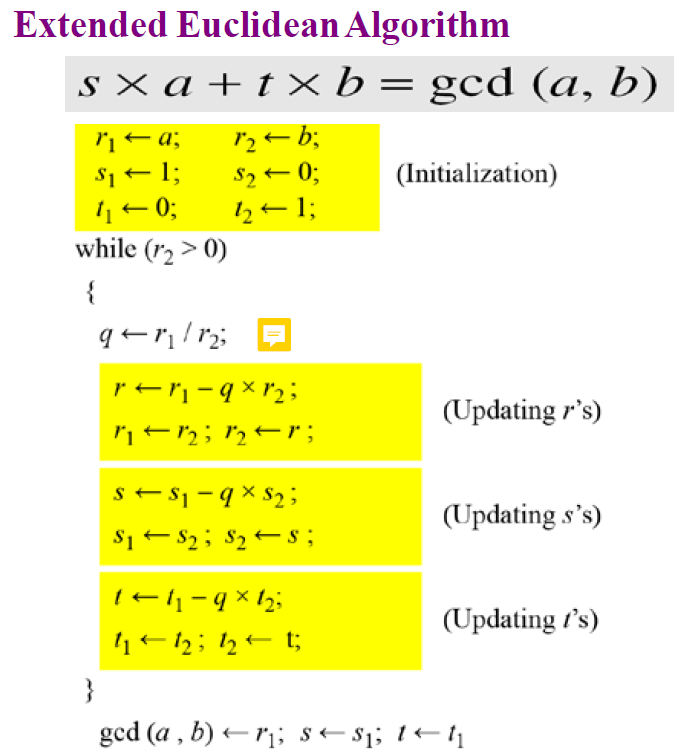
整除：n | a, 不能整除：𝒏∤𝒂



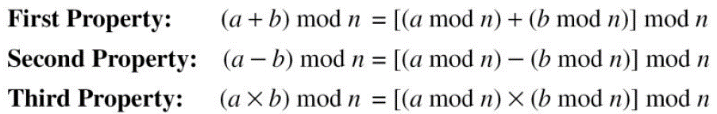


When gcd (a, b) = 1, we say that a and b are relatively prim.





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





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

Inverse:

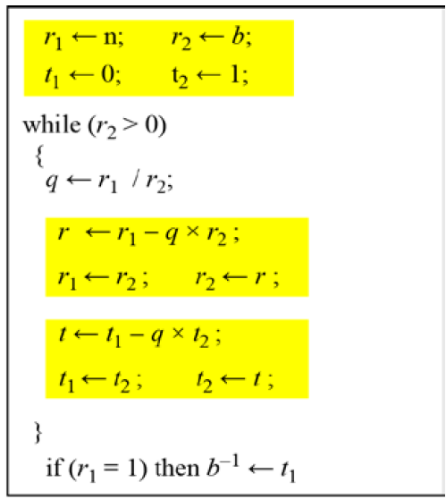
Additive Inverse:

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

Multiplicative Inverse:

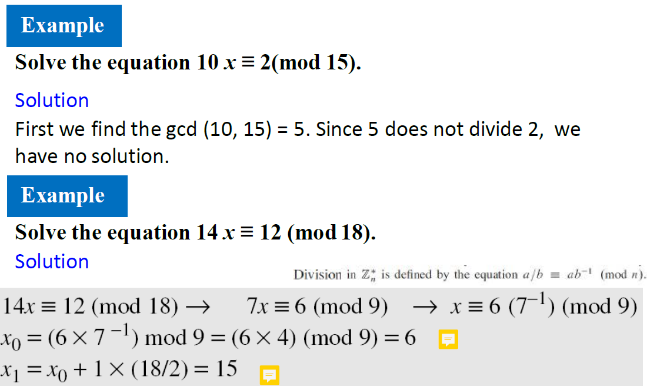
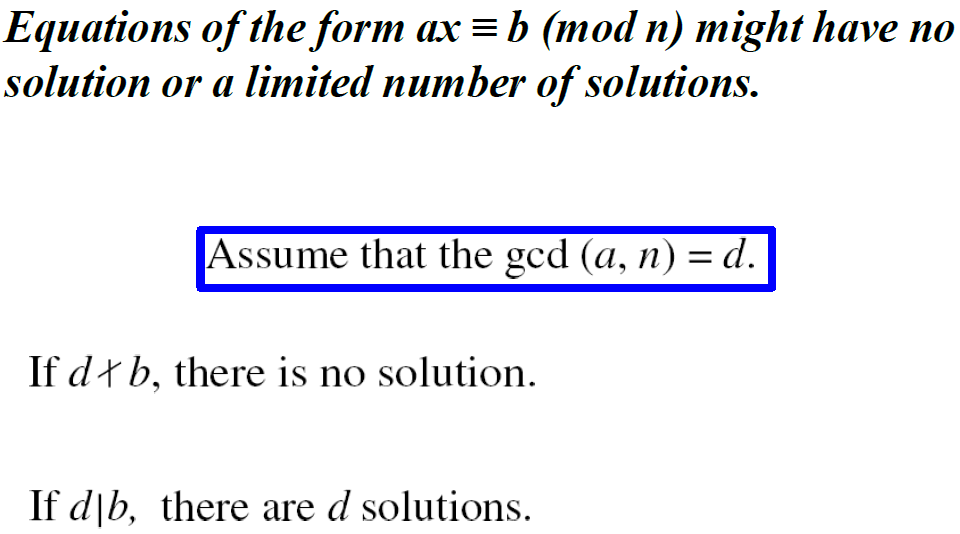
In Zn, two numbers a and b are the multiplicative inverse of each other if: a x b ≡ 1 (mod n), a = b-1

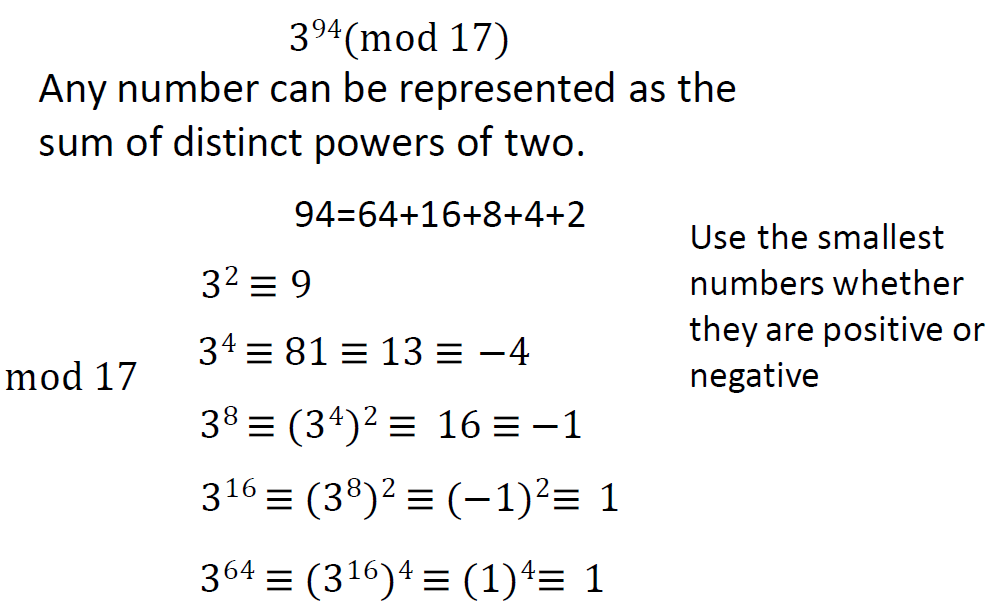
no multiplicative inverse if gcd (10, 8) = 2 ≠ 1.

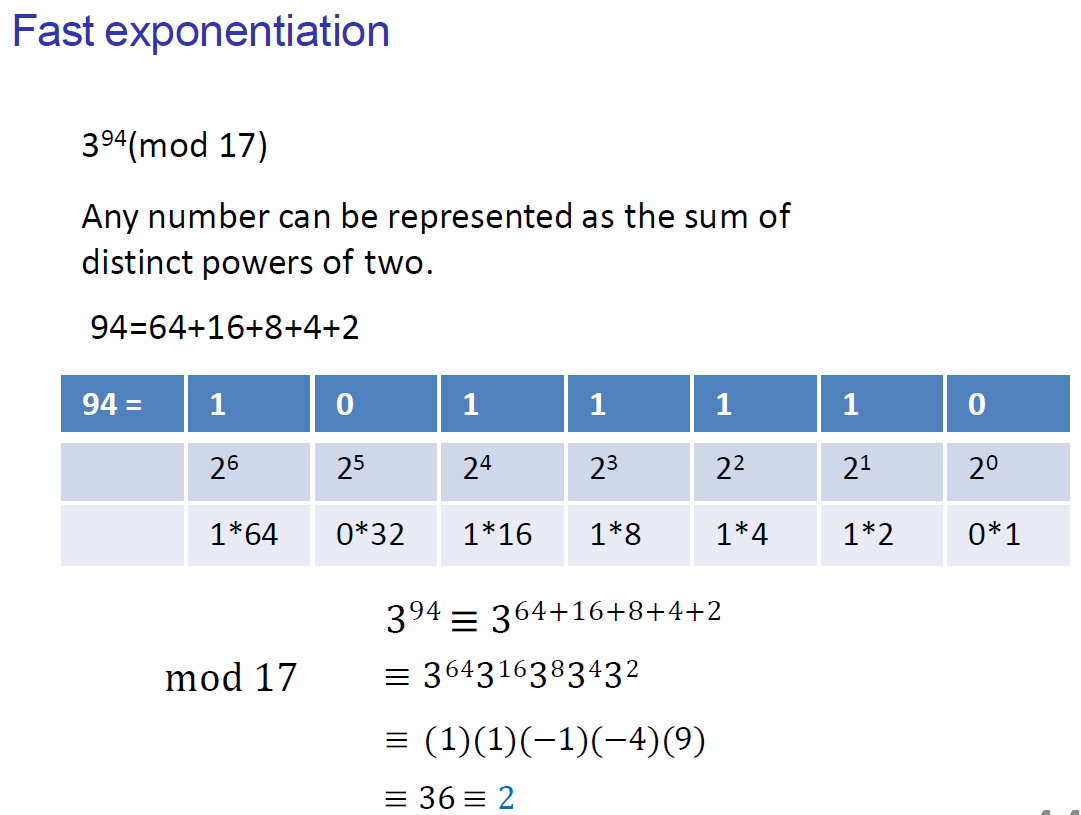


Zn\*: Zn 中所有和 n 互质的数 (gcd(a,n)=1)

single-Variable Linear Equations



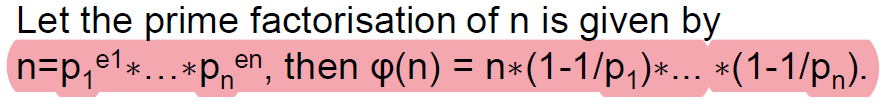


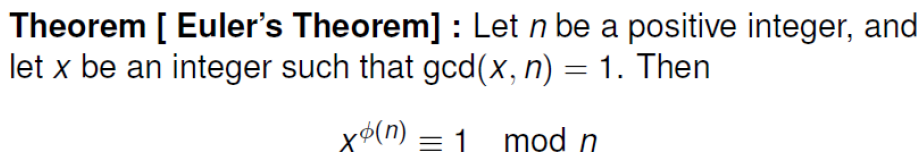




Euler’s function

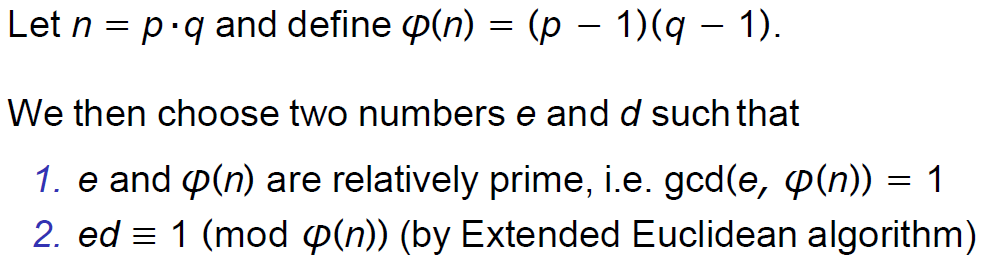
φ(n) 是 Zn\* 的长度

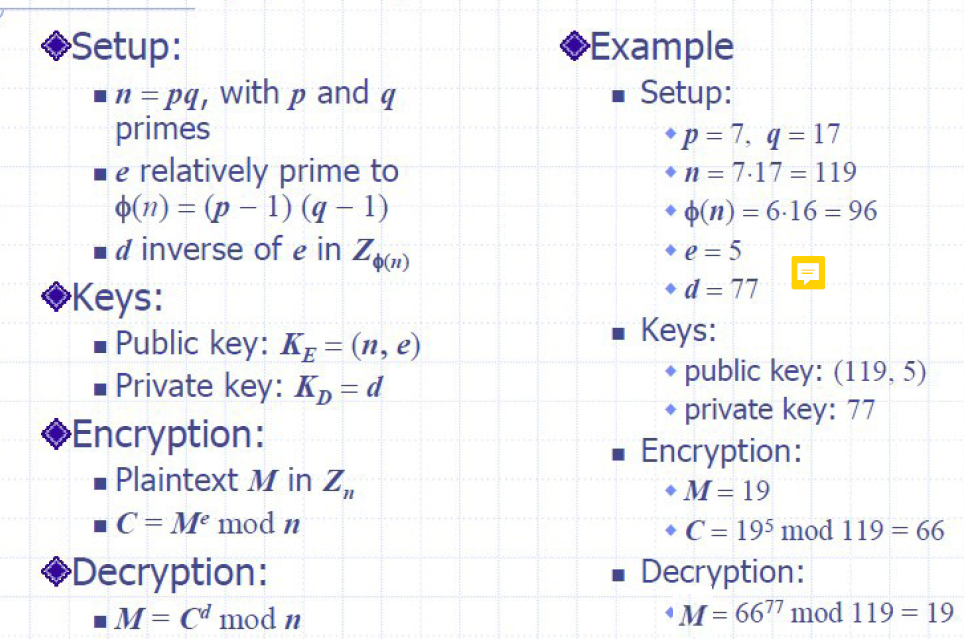




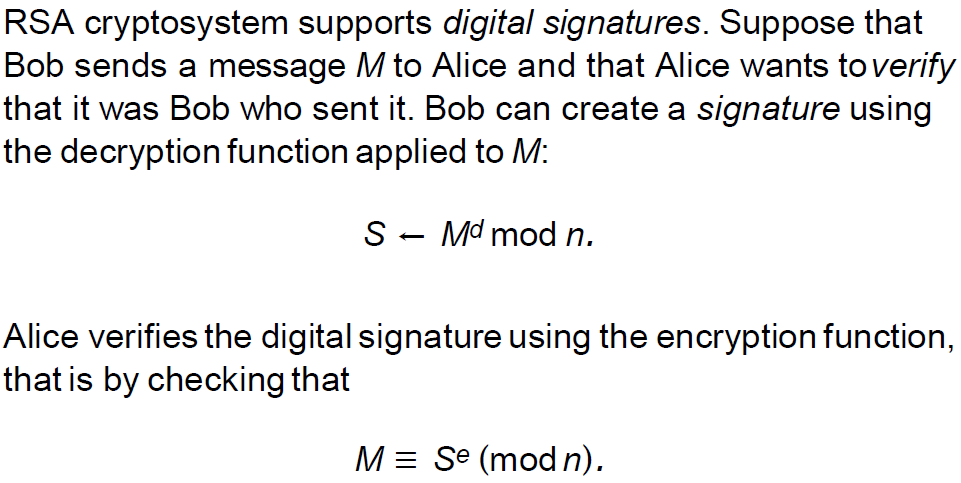
plaintext: 明文, ciphertext: 加密文

28, RSA encryption scheme:





29, Digital signatures:



P ⊆ N P

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 input s 可以通过一个函数 f(s) 转变到 M 的 input 里，那么 L 就可以被约化到 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:



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

Triangle Inequality TSP:

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

