# Trie,Border 树

# 问题背景

## 背景

**字典**：一个字符串的集合称为字典。
**字典串**：在字典里的串称为字典串。
在处理字符串的时候，常会遇到这样的简单问题：给出一个字典，然后回答大量询问：输入一个字符串，判断它是否在字典中。

# Trie

## 定义

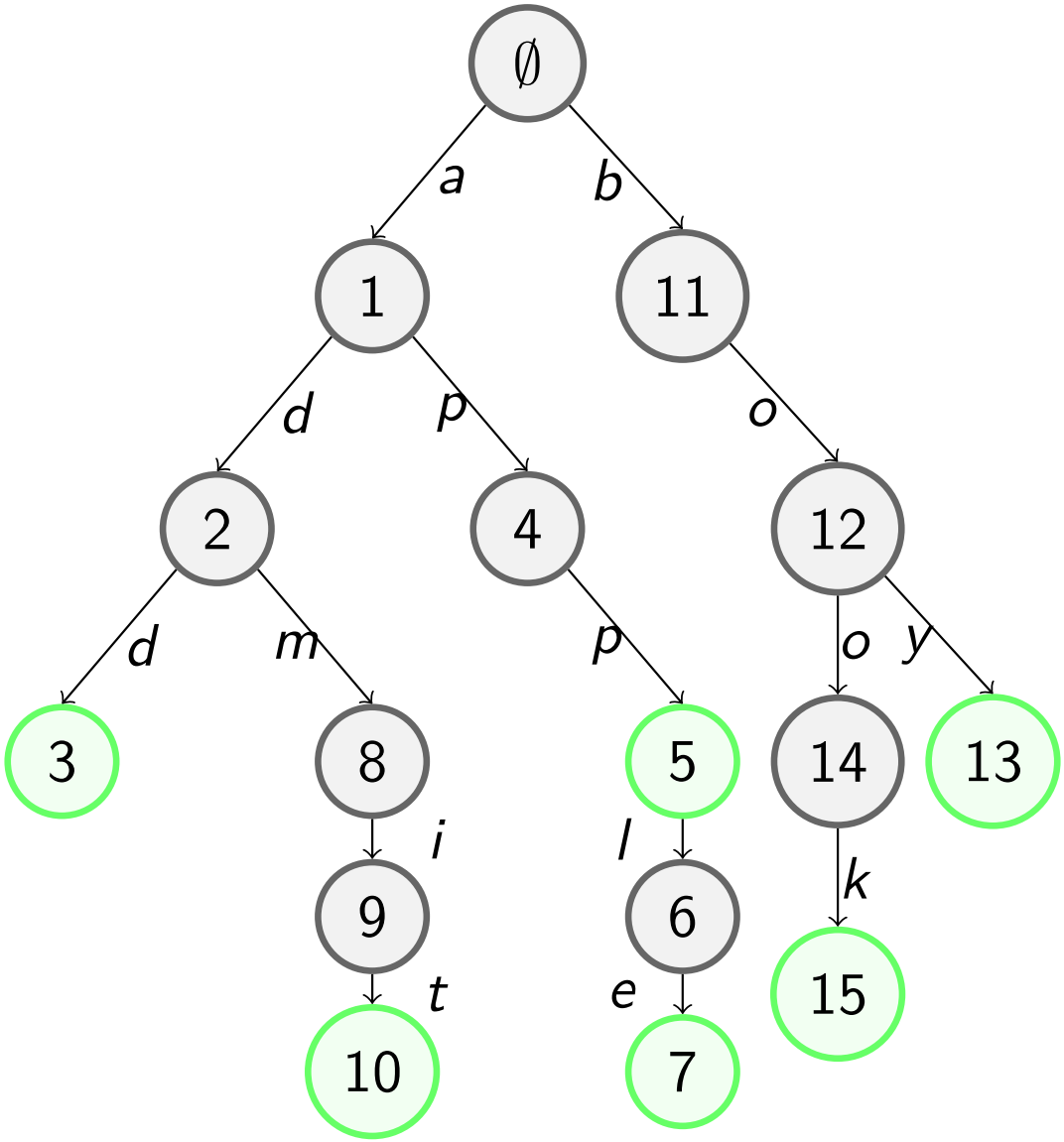
Trie 是一棵有根树，每个点至多有 $|\Sigma|$ 个后继边，每条边上有一个字符。
每个点表示一个前缀：从跟到这个点的边上的字符顺次连接形成的字符串。
每个点还有一个终止标记：是否这个点代表的字符串是一个字典串。

# Trie

## 定义

Trie 是一棵有根树，每个点至多有 $|\Sigma|$ 个后继边，每条边上有一个字符。
每个点表示一个前缀：从跟到这个点的边上的字符顺次连接形成的字符串。
每个点还有一个终止标记：是否这个点代表的字符串是一个字典串。

可以支持向 Trie 插入新字典串，删除字典串，查询某字符串是否是字典串，以及一些稍微复杂的查询。
作为一个数据结构，它理所应当的还可以进行持久化。


∅
a
b
1
11
d
p
o
2
4
12
d
m
p
o
y
3
8
5
14
13
i
l
k
9
6
15
t
e
10
7

# 例题 1-Trie

## 练习赛 11-假的字符串

给定 $n$ 个不同的字符串 $S_1, S_2, \cdots, S_n$ 你可以任意指定字符之间的大小关系（即重定义字典序），求有多少个串可能成为字典序最小的串。

$n \leq 30,000 \quad \sum_{i=1}^{n} |S_i| \leq 300,000$

# 例题 1-Trie

## 练习赛 11-假的字符串

给定 $n$ 个不同的字符串 $S_1, S_2, \cdots, S_n$ 你可以任意指定字符之间的大小关系（即重定义字典序），求有多少个串可能成为字典序最小的串。
$n \leq 30,000 \quad \sum_{i=1}^{n} |S_i| \leq 300,000$

## 题解

先构建出字典树，然后逐个字符串判定可行性。

# 例题 1-Trie

## 练习赛 11-假的字符串

给定 $n$ 个不同的字符串 $S_1, S_2, \cdots, S_n$ 你可以任意指定字符之间的大小关系（即重定义字典序），求有多少个串可能成为字典序最小的串。
$n \leq 30,000 \quad \sum_{i=1}^{n} |S_i| \leq 300,000$

## 题解

先构建出字典树，然后逐个字符串判定可行性。
考虑 $S_i$ 在字典树上的每个节点，如果有多于一个后继边，则 $S_i$ 使用的字母必须小于其他字母。等价于判定 26 个点的有向图上是否有环（拓扑排序）。

# 例题 1-Trie

## 练习赛 11-假的字符串

给定 $n$ 个不同的字符串 $S_1, S_2, \cdots, S_n$ 你可以任意指定字符之间的大小关系（即重定义字典序），求有多少个串可能成为字典序最小的串。
$n \leq 30,000 \quad \sum_{i=1}^{n} |S_i| \leq 300,000$

## 题解

先构建出字典树，然后逐个字符串判定可行性。
考虑 $S_i$ 在字典树上的每个节点，如果有多于一个后继边，则 $S_i$ 使用的字母必须小于其他字母。等价于判定 26 个点的有向图上是否有环（拓扑排序）。
额外需要注意不能有任何其他串等于 $S_i$ 的前缀，即路径上不能有其他串的的终止节点。

# 01-Trie

## 定义

01-Trie 即对字符集只有 2 的字典串构建的 Trie。常来解决 01 二进制串相关的字典序或者计数问题。

# 例题 2-01-Trie

## 奶牛异或

给出一个正整数数组 $A$，长度不超过 $100,000$。定义区间异或和为区间所有数字异或起来的结果。
求最大区间异或和。

# 例题 2-01-Trie

## 奶牛异或

给出一个正整数数组 $A$，长度不超过 $100,000$。定义区间异或和为区间所有数字异或起来的结果。
求最大区间异或和。

## 题解

首先求出异或前缀和
$Sum[0] = 0, Sum[1] = A[1], Sum[2] = Sum[1] \oplus A[2], \cdots, Sum[n] = Sum[n-1] \oplus A[n]$
则每个区间的异或和可以表示成两个前缀和的异或。问题转化为 $n+1$ 个数字中选两个做异或运算的最大值。可以用 01Trie 解决。

# 例题 2-01-Trie

## 题解

枚举一个数字 $x$，然后寻找与他异或结果最大的另一个数字 $y$。

# 例题 2-01-Trie

## 题解

枚举一个数字 $x$，然后寻找与他异或结果最大的另一个数字 $y$。
由于相同宽度的两个二进制数字的大小关系等价于字典序关系。

# 例题 2-01-Trie

## 题解

枚举一个数字 $x$，然后寻找与他异或结果最大的另一个数字 $y$。
由于相同宽度的两个二进制数字的大小关系等价于字典序关系。
从高到低考虑 $x$ 的每一个二进制位 $bit$：如果 $y$ 的这一位也是 $bit$，则异或结果的这一位为 $0$；如果 $y$ 的这一位是 $!bit$，则异或结果的这一位为 $1$。

# 例题 2-01-Trie

## 题解

枚举一个数字 $x$，然后寻找与他异或结果最大的另一个数字 $y$。
由于相同宽度的两个二进制数字的大小关系等价于字典序关系。
从高到低考虑 $x$ 的每一个二进制位 $bit$：如果 $y$ 的这一位也是 $bit$，则异或结果的这一位为 $0$；如果 $y$ 的这一位是 $!bit$，则异或结果的这一位为 $1$。
将所有数字插入到 01Trie 中，枚举 $x$，在 01Trie 上寻找 $y$：从根出发，如果有 $!bit$ 边，则走 $!bit$ 边，否则只能走 $bit$ 边。

# 例题 3-01-Trie

## 某谷 P4735 & 牛客 51120

给出一个数组 $A$，长度为 $n \leq 300,000$，再给出 $Q \leq 300,000$ 次操作:
1. 修改操作：将数字 $x$ 添加到数组末尾，数组长度 $n$ 变为 $n+1$。
2. 查询操作：求区间 $[\mathbf{l}, \mathbf{r}]$ 内的 $p$，使得 $A[p] \oplus A[p+1] \oplus \cdots \oplus A[n] \oplus \mathbf{X}$ 最大。

# 例题 3-01-Trie

## 某谷 P4735 & 牛客 51120

给出一个数组 $A$，长度为 $n \leq 300,000$，再给出 $Q \leq 300,000$ 次操作:
1. 修改操作：将数字 $x$ 添加到数组末尾，数组长度 $n$ 变为 $n+1$。
2. 查询操作：求区间 $[\mathbf{l}, \mathbf{r}]$ 内的 $p$，使得 $A[p] \oplus A[p+1] \oplus \cdots \oplus A[n] \oplus \mathbf{X}$ 最大。

## 题解

从查询入手，

$$A[p] \oplus A[p+1] \oplus \cdots \oplus A[n] \oplus X = Sum[n] \oplus Sum[p-1] \oplus X$$

# 例题 3-01-Trie

## 某谷 P4735 & 牛客 51120

给出一个数组 $A$，长度为 $n \leq 300,000$，再给出 $Q \leq 300,000$ 次操作:
1. 修改操作：将数字 $x$ 添加到数组末尾，数组长度 $n$ 变为 $n+1$。
2. 查询操作：求区间 $[\mathbf{l}, \mathbf{r}]$ 内的 $p$，使得 $A[p] \oplus A[p+1] \oplus \cdots \oplus A[n] \oplus \mathbf{X}$ 最大。

## 题解

从查询入手，

$$A[p] \oplus A[p+1] \oplus \cdots \oplus A[n] \oplus X = Sum[n] \oplus Sum[p-1] \oplus X$$

由于 $Y = Sum[n] \oplus X$ 为定值，因此查询等价于求最大的 $Sum[p-1] \oplus Y$。

# 例题 3-01-Trie

## 某谷 P4735 & 牛客 51120

给出一个数组 $A$，长度为 $n \leq 300,000$，再给出 $Q \leq 300,000$ 次操作:
1. 修改操作：将数字 $x$ 添加到数组末尾，数组长度 $n$ 变为 $n+1$。
2. 查询操作：求区间 $[\mathbf{l}, \mathbf{r}]$ 内的 $p$，使得 $A[p] \oplus A[p+1] \oplus \cdots \oplus A[n] \oplus \mathbf{X}$ 最大。

## 题解

从查询入手，

$$A[p] \oplus A[p+1] \oplus \cdots \oplus A[n] \oplus X = Sum[n] \oplus Sum[p-1] \oplus X$$

由于 $Y = Sum[n] \oplus X$ 为定值，因此查询等价于求最大的 $Sum[p-1] \oplus Y$。
如果没有限制 $p$ 的范围，则用 01-Trie 解决即可。
本题限制了 $p$ 的区间，则用持久化 01-Trie 解决即可。

# 例题 4-01-Trie

## CF 888G Xor-MST & 牛客 112412

有一个 $n \leq 200,000$ 的无向完全图，每个点 $u$ 有一个点权为 $0 \leq A[u] \leq 2^{30}$。
任意两个点 $u, v$ 之间的边权为 $A[u] \oplus A[v]$。求最小生成树。

# 例题 4-01-Trie

## CF 888G Xor-MST & 牛客 112412

有一个 $n \leq 200,000$ 的无向完全图，每个点 $u$ 有一个点权为 $0 \leq A[u] \leq 2^{30}$。
任意两个点 $u, v$ 之间的边权为 $A[u] \oplus A[v]$。求最小生成树。

## 题解

较常见的两种生成树算法 *Kruskal* 和 *Prim* 都不能很好的处理本题。

# 例题 4-01-Trie

## CF 888G Xor-MST & 牛客 112412

有一个 $n \leq 200,000$ 的无向完全图，每个点 $u$ 有一个点权为 $0 \leq A[u] \leq 2^{30}$。
任意两个点 $u, v$ 之间的边权为 $A[u] \oplus A[v]$。求最小生成树。

## 题解

较常见的两种生成树算法 *Kruskal* 和 *Prim* 都不能很好的处理本题。
江湖中流传着另一种生成树算法 Boruvka：每一轮每个连通块独立去寻找连接另一个连通块的最短边，如此的迭代至多进行 *logn* 轮。

# 例题 4-01-Trie

## CF 888G Xor-MST & 牛客 112412

有一个 $n \leq 200,000$ 的无向完全图，每个点 $u$ 有一个点权为 $0 \leq A[u] \leq 2^{30}$。
任意两个点 $u, v$ 之间的边权为 $A[u] \oplus A[v]$。求最小生成树。

## 题解

较常见的两种生成树算法 *Kruskal* 和 *Prim* 都不能很好的处理本题。
江湖中流传着另一种生成树算法 Boruvka：每一轮每个连通块独立去寻找连接另一个连通块的最短边，如此的迭代至多进行 *logn* 轮。
先将所有 $A[i]$ 插入 01-Trie。在每轮迭代中，枚举所有连通块，枚举连通块里的值 $A[j]$，求与其他连通块内点的最小异或值。可以先将该连通块的点先从 01-Trie 中删除，在处理完该连通块之后，再将他们放回。

# Border 树

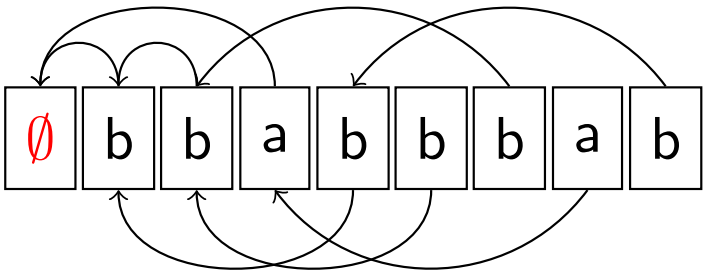
## 定义

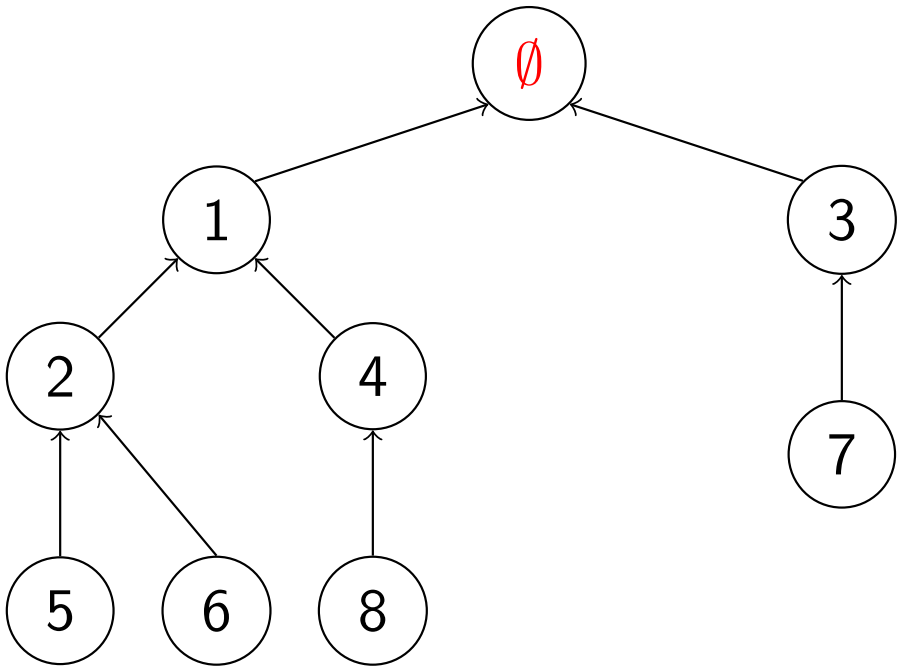
**对于一个字符串 $S$，$n = |S|$，它的 Border 树（也叫 next 树）共有 $n+1$ 个节点：$0, 1, 2, \ldots, n$。**
**0 是这棵有向树的根。对于其他每个点 $1 \leq i \leq n$，父节点为 $next[i]$。**

# Border 树

## 定义

**对于一个字符串 $S$，$n = |S|$，它的 Border 树（也叫 next 树）共有 $n+1$ 个节点：$0, 1, 2, \ldots, n$。**
**0 是这棵有向树的根。对于其他每个点 $1 \le i \le n$，父节点为 $next[i]$。**



# Border 树

## 性质

1. 每个前缀 $Prefix[i]$ 的所有 Border：节点 $i$ 到根的链。
2. 哪些前缀有长度为 $x$ 的 Border：$x$ 的子树。
3. 求两个前缀的公共 Border 等价于求 LCA。

# Border 树

## 性质

1. 每个前缀 $Prefix[i]$ 的所有 Border：节点 $i$ 到根的链。
2. 哪些前缀有长度为 $x$ 的 Border：$x$ 的子树。
3. 求两个前缀的公共 Border 等价于求 LCA。

## 例题 5-Border 树 某谷 P5829

给出一个字符串 $S$，$|S| \leq 1000,000$，有 $Q \leq 100,000$ 次询问：前缀 $S[1,p]$ 与前缀 $S[1,q]$ 的最大公共 Border 长度。

# Border 树

## 例题 6-Border 树

字符串 $S$ 长度不超过 $10^6$，求一个最长的子串 $T$，满足：

- $T$ 为 $S$ 的前缀。
- $T$ 为 $S$ 的后缀。
- $T$ 在 S 中至少出现$K$次。

# Border 树

## 例题 6-Border 树

字符串 $S$ 长度不超过 $10^6$，求一个最长的子串 $T$，满足：

- $T$ 为 $S$ 的前缀。
- $T$ 为 $S$ 的后缀。
- $T$ 在 S 中至少出现$K$次。

## 例题 7-Border 树

字符串 $S$ 长度不超过 $200,000$，有 $Q \leq 200,000$ 次操作：

1. 修改操作：向字符串末尾添加一个字符 $ch$。
2. 查询操作：求一个最长的子串 $T$，满足：
   - $T$ 为 $S$ 的前缀。
   - $T$ 为 $S$ 的后缀。
   - $T$ 在 S 中至少出现$K$次。

# AC 自动机

AC 自动机 = Trie + Border 树。
日后再讲。