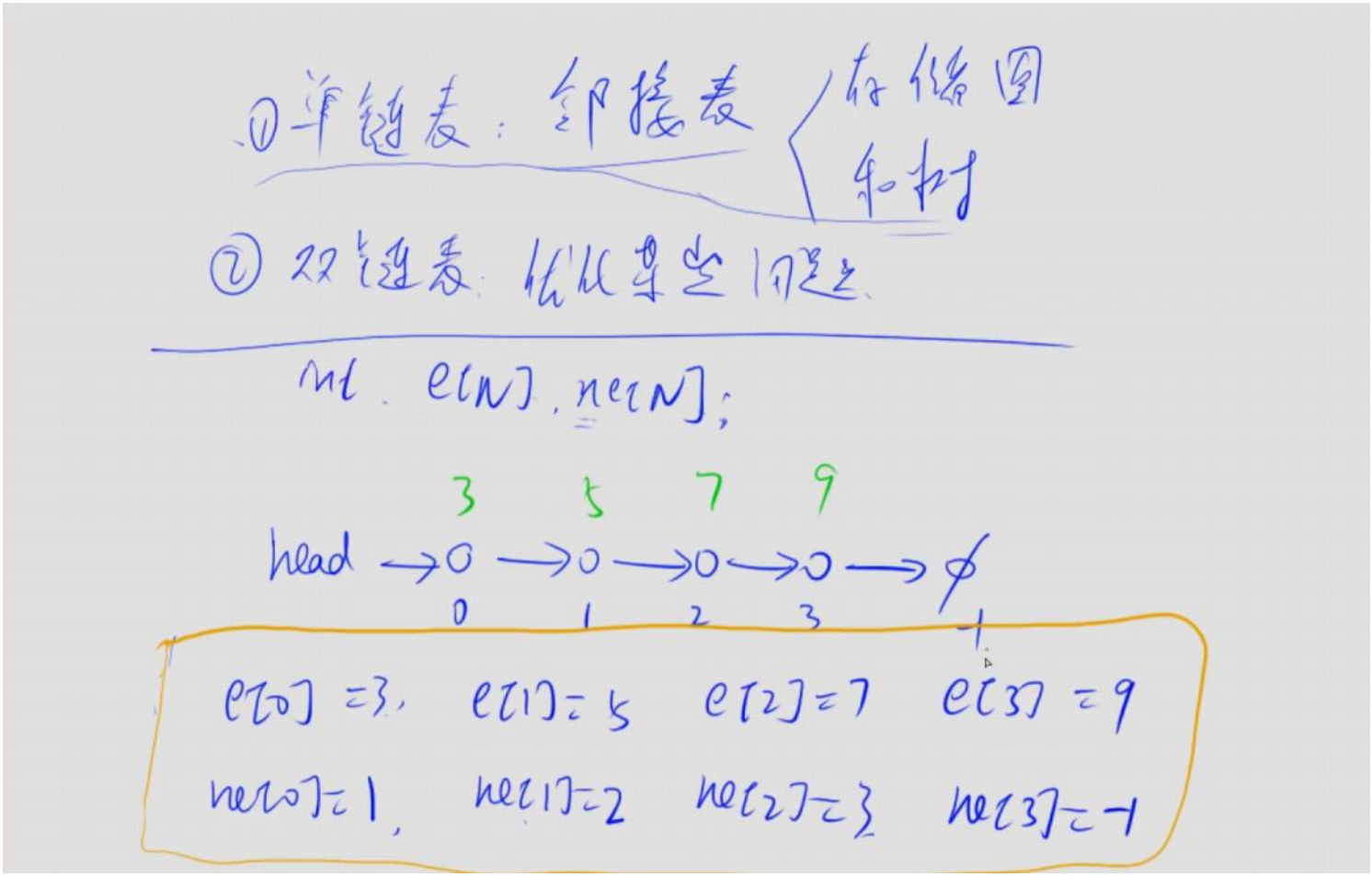
链表

2024年1月17日 4:04



单链表

2024年2月22日 20:02

题目 提交记录 讨论 题解

实现一个单链表，链表初始为空，支持三种操作：

(1) 向链表头插入一个数；

(2) 删除第k个插入的数后面的数；

(3) 在第k个插入的数后插入一个数

现在要对该链表进行M次操作，进行完所有操作后，从头到尾输出整个链表。

注意:题目中第k个插入的数并不是指当前链表的第k个数。例如操作过程中一共插入了n个数，则按照插入的时间顺序，这n个数依次为：第1个插入的数，第2个插入的数，…第n个插入的数。

输入格式

第一行包含整数M，表示操作次数。

接下来M行，每行包含一个操作命令，操作命令可能为以下几种：

(1) “H x”，表示向链表头插入一个数x。

(2) “D k”，表示删除第k个输入的数后面的数（当k为0时，表示删除头结点）。

(3) “I k x”，表示在第k个输入的数后面插入一个数x（此操作中k均大于0）。

输出格式

共一行，将整个链表从头到尾输出。

数据范围

$1 \leq M \leq 100000$
所有操作保证合法。

输入样例:

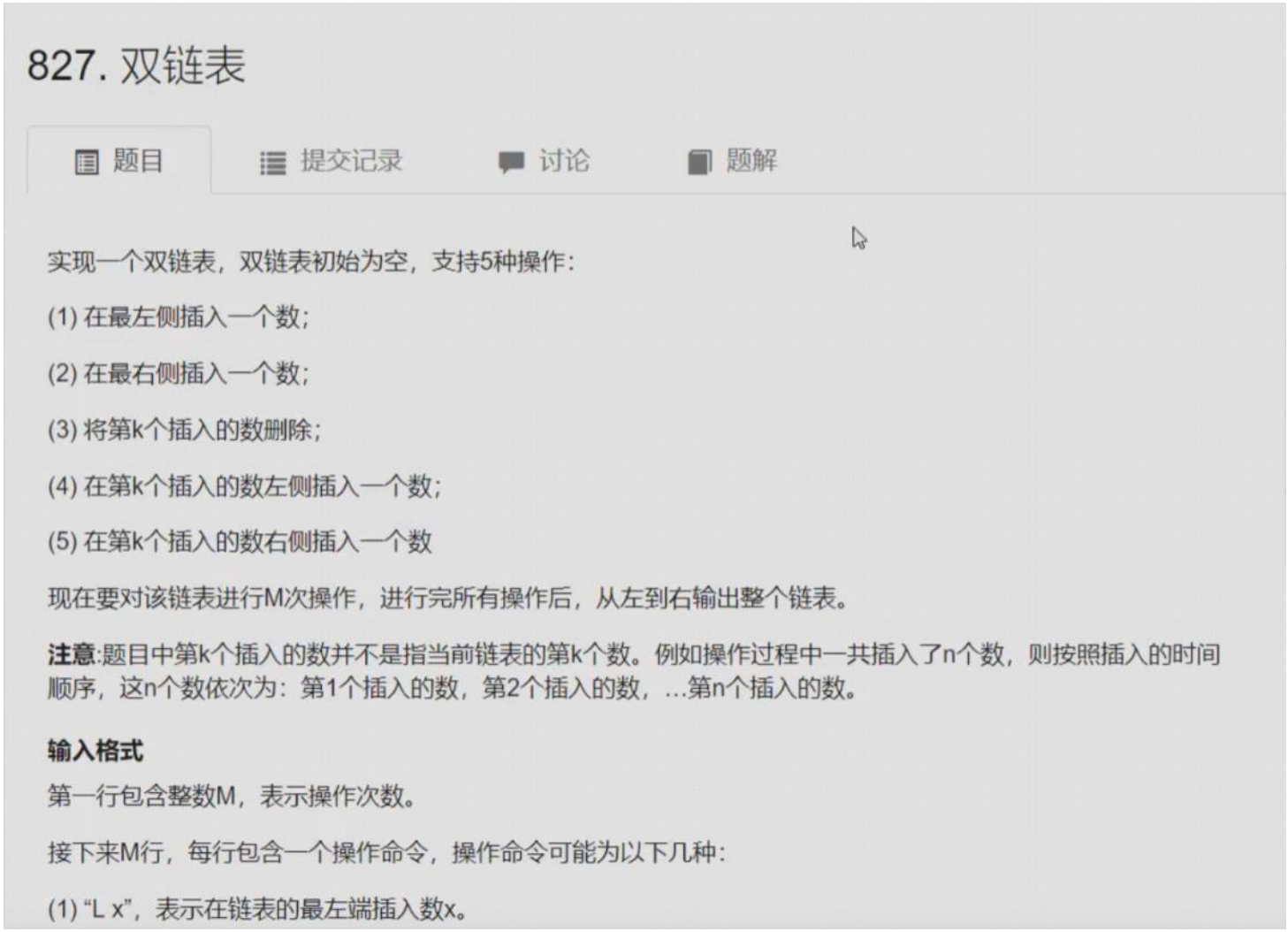
```
10
H 9
I 1 1
D 1
D 0
H 6
I 3 6
I 4 5
I 4 5
I 3 4
D 6
```

输出样例:

```
6 4 6 5
```

双链表

2024年2月22日 20:02



827. 双链表

题目 提交记录 讨论 题解

实现一个双链表，双链表初始为空，支持5种操作：

(1) 在最左侧插入一个数；

(2) 在最右侧插入一个数；

(3) 将第k个插入的数删除；

(4) 在第k个插入的数左侧插入一个数；

(5) 在第k个插入的数右侧插入一个数

现在要对该链表进行M次操作，进行完所有操作后，从左到右输出整个链表。

注意:题目中第k个插入的数并不是指当前链表的第k个数。例如操作过程中一共插入了n个数，则按照插入的时间顺序，这n个数依次为：第1个插入的数，第2个插入的数，…第n个插入的数。

输入格式

第一行包含整数M，表示操作次数。

接下来M行，每行包含一个操作命令，操作命令可能为以下几种：

(1) “L x”，表示在链表的最左端插入数x。

(2) “R x”，表示在链表的最右端插入数x。

(3) “D k”，表示将第k个插入的数删除。

(4) “IL k x”，表示在第k个插入的数左侧插入一个数。

(5) “IR k x”，表示在第k个插入的数右侧插入一个数。

输出格式

共一行，将整个链表从左到右输出。

数据范围

$1 \le M \le 100000$
所有操作保证合法。

输入样例：

```
10
R 7
D 1
L 3
IL 2 10
D 3
IL 2 7
L 8
R 9
IL 4 7
IR 2 2
```

输出样例：

```
8 7 7 3 2 9
```

栈

2024年2月22日 20:03

```cpp
#include <iostream>

using namespace std;

const int N = 100010;

// ********************** 栈
int stk[N], tt;

// 插入
stk[ ++ tt] = x;

// 弹出
tt --;

// 判断栈是否为空
if (tt > 0) not empty
else empty

// 栈顶
stk[tt];

// ********************** 队列

// 在队尾插入元素，在队头弹出元素
int q[N], hh, tt = -1;

// 插入
q[ ++ tt] = x;

// 弹出
hh ++ ;

```

模拟栈

2024年2月22日 20:03

828. 模拟栈

题目 提交记录 讨论 题解

实现一个栈，栈初始为空，支持四种操作：

(1) “push x” – 向栈顶插入一个数x；

(2) “pop” – 从栈顶弹出一个数；

(3) “empty” – 判断栈是否为空；

(4) “query” – 查询栈顶元素。

现在要对栈进行M个操作，其中的每个操作3和操作4都要输出相应的结果。

输入格式

第一行包含整数M，表示操作次数。

接下来M行，每行包含一个操作命令，操作命令为”push x”，”pop”，”empty”，”query”中的一种。

输出格式

对于每个”empty”和”query”操作都要输出一个查询结果，每个结果占一行。

其中，”empty”操作的查询结果为“YES”或“NO”，”query”操作的查询结果为一个整数，表示栈顶元素的值。

数据范围

$1 \le M \le 100000$,
$1 \le x \le 10^9$
所有操作保证合法。

输入样例：

```
10
push 5
query
push 6
pop
query
pop
empty
push 4
query
empty
```

输出样例：

```
5
5
YES
4
NO
```

单调栈

2024年2月22日 20:03

给定一个长度为N的整数数列，输出每个数左边第一个比它小的数，如果不存在则输出-1。

输入格式

第一行包含整数N，表示数列长度。

第二行包含N个整数，表示整数数列。

输出格式

共一行，包含N个整数，其中第i个数表示第i个数的左边第一个比它小的数，如果不存在则输出-1。

数据范围

$1 \leq N \leq 10^5$
$1 \leq$ 数列中元素 $\leq 10^9$

输入样例：

```
5
3 4 2 7 5
```

输出样例：

```
-1 3 -1 2 2
```

队列

2024年2月22日 20:03

```cpp
#include <iostream>

using namespace std;

const int N = 100010;

// ********************** 栈
int stk[N], tt;

// 插入
stk[ ++ tt] = x;

// 弹出
tt --;

// 判断栈是否为空
if (tt > 0) not empty
else empty

// 栈顶
stk[tt];

// ********************** 队列

// 在队尾插入元素，在队头弹出元素
int q[N], hh, tt = -1;

// 插入
q[ ++ tt] = x;

// 弹出
hh ++ ;

```

模拟队列

2024年2月22日 20:03

单调队列

2024年2月22日 20:15

154. 滑动窗口

题目 提交记录 讨论 题解

给定一个大小为$n \leq 10^6$的数组。

有一个大小为k的滑动窗口，它从数组的最左边移动到最右边。

您只能在窗口中看到k个数字。

每次滑动窗口向右移动一个位置。

以下是一个例子：

该数组为[1 3 -1 -3 5 3 6 7]，k为3。

窗口位置	最小值	最大值
[1 3 -1] -3 5 3 6 7	-1	-3
1 [3 -1 -3] 5 3 6 7	-3	3
1 3 [-1 -3 5] 3 6 7	-3	5
1 3 -1 [-3 5 3] 6 7	-3	5
1 3 -1 -3 [5 3 6] 7	3	6
1 3 -1 -3 5 [3 6 7]	3	7

您的任务是确定滑动窗口位于每个位置时，窗口中的最大值和最小值。

输入格式

输入包含两行。

第一行包含两个整数n和k，分别代表数组长度和滑动窗口的长度。

第二行有n个整数，代表数组的具体数值。

同行数据之间用空格隔开。

输出格式

输出包含两个。

第一行输出，从左至右，每个位置滑动窗口中的最小值。

第二行输出，从左至右，每个位置滑动窗口中的最大值。

输入样例:

```
8 3
1 3 -1 -3 5 3 6 7
```

输出样例:

```
-1 -3 -3 -3 3 3
3 3 5 5 6 7
```

KMP

2024年2月22日 20:04

831. KMP字符串

题目 提交记录 讨论 题解

给定一个模式串S，以及一个模板串P，所有字符串中只包含大小写英文字母以及阿拉伯数字。

模板串P在模式串S中多次作为子串出现。

求出模板串P在模式串S中所有出现的位置的起始下标。

输入格式

第一行输入整数N，表示字符串P的长度。

第二行输入字符串P。

第三行输入整数M，表示字符串S的长度。

第四行输入字符串M。

输出格式

共一行，输出所有出现位置的起始下标（下标从0开始计数），整数之间用空格隔开。

数据范围

$1 \leq N \leq 10^4$
$1 \leq M \leq 10^5$

输入样例：

```
3
aba
5
ababa
```

输出样例：

```
0 2
```

Trie树

2024年2月22日 20:04

835. Trie字符串统计

题目 | 提交记录 | 讨论 | 题解

维护一个字符串集合，支持两种操作：

1. “I x”向集合中插入一个字符串x；
2. “Q x”询问一个字符串在集合中出现了多少次。

共有N个操作，输入的字符串总长度不超过 10^5，字符串仅包含小写英文字母。

输入格式

第一行包含整数N，表示操作数。

接下来N行，每行包含一个操作指令，指令为”I x”或”Q x”中的一种。

输出格式

对于每个询问指令”Q x”，都要输出一个整数作为结果，表示x在集合中出现的次数。

每个结果占一行。

数据范围

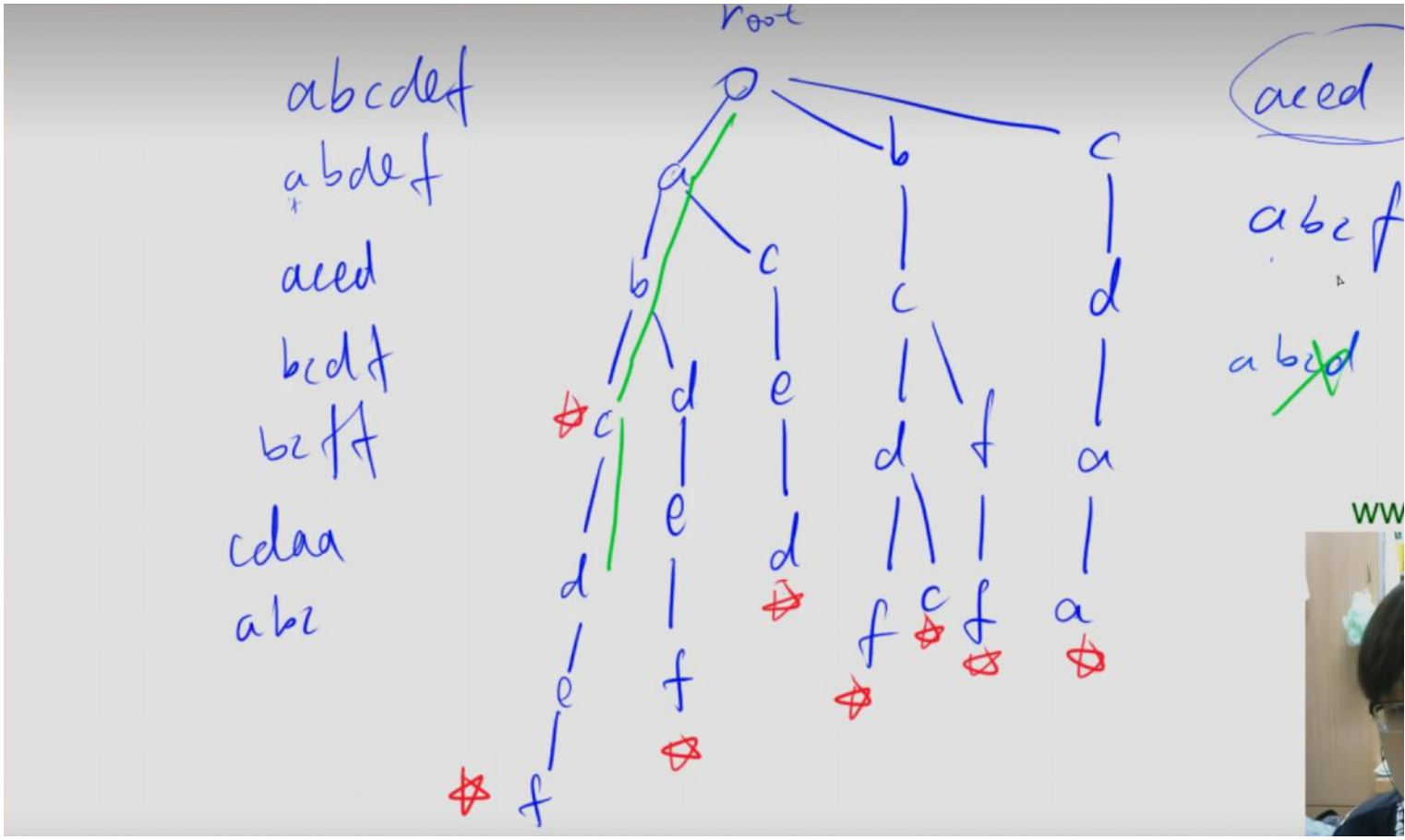
$1 \leq N \leq 2 * 10^4$

输入样例：

```
5
I abc
Q abc
Q ab
I ab
Q ab
```

输出样例：

```
1
0
1
```



快速的存取字符串集合

并查集

2024年2月22日 20:04

朴素并查集

2024年2月22日 20:05

836. 合并集合

题目 提交记录 讨论 题解

一共有n个数，编号是1~n，最开始每个数各自在一个集合中。

现在要进行m个操作，操作共有两种：

1. “M a b”，将编号为a和b的两个数所在的集合合并，如果两个数已经在同一个集合中，则忽略这个操作；
2. “Q a b”，询问编号为a和b的两个数是否在同一个集合中；

输入格式

第一行输入整数n和m。

接下来m行，每行包含一个操作指令，指令为“M a b”或“Q a b”中的一种。

输出格式

对于每个询问指令”Q a b”，都要输出一个结果，如果a和b在同一集合内，则输出“Yes”，否则输出“No”。

每个结果占一行。

数据范围

$1 \leq n, m \leq 10^5$

输入样例：

```
4 5
M 1 2
M 3 4
Q 1 2
Q 1 3
Q 3 4
```

输出样例：

```
Yes
No
Yes
```

并查集：

1. 将两个集合合并
2. 询问两个元素是否在一个集合当中

基本原理：每个集合用一棵树来表示。树根的编号就是整个集合的编号。每个节点存储它的父节点，p[x]表示x的父节点

问题1：如何判断树根：if (p[x] == x)

问题2：如何求x的集合编号：while (p[x] != x) x = p[x];

问题3：如何合并两个集合：px 是x的集合编号，py是y的集合编号。p[x] = y

维护size的并查集

2024年2月22日 20:05

837. 连通块中点的数量

题目 提交记录 讨论 题解

给定一个包含n个点（编号为1~n）的无向图，初始时图中没有边。

现在要进行m个操作，操作共有三种：

1. “C a b”，在点a和点b之间连一条边，a和b可能相等；
2. “Q1 a b”，询问点a和点b是否在同一个连通块中，a和b可能相等；
3. “Q2 a”，询问点a所在连通块中点的数量；

输入格式

第一行输入整数n和m。

接下来m行，每行包含一个操作指令，指令为“C a b”，“Q1 a b”或“Q2 a”中的一种。

输出格式

对于每个询问指令”Q1 a b”，如果a和b在同一个连通块中，则输出“Yes”，否则输出“No”。

对于每个询问指令“Q2 a”，输出一个整数表示点a所在连通块中点的数量

每个结果占一行。

数据范围

$1 \leq n, m \leq 10^5$

输入样例：

```
5 5
C 1 2
Q1 1 2
Q2 1
C 2 5
Q2 5
```

输出样例：

```
Yes
2
3
```

维护到祖宗节点距离的并查集

2024年2月22日 20:05

堆

2024年2月22日 20:05

堆排序

2024年2月22日 20:31

838. 堆排序

题目 | 提交记录 | 讨论 | 题解

输入一个长度为n的整数数列，从小到大输出前m小的数。

输入格式

第一行包含整数n和m。

第二行包含n个整数，表示整数数列。

输出格式

共一行，包含m个整数，表示整数数列中前m小的数。

数据范围

$1 \leq m \leq n \leq 10^5$,
$1 \leq$ 数列中元素 $\leq 10^9$

输入样例:

```
5 3
4 5 1 3 2
```

输出样例:

```
1 2 3
```

如何手写一个堆?

1. 插入一个数 — `heap[ ++ size] = x; up(size);`
2. 求集合当中的最小值 — `heap[1];`
3. 删除最小值 — `heap[1] = heap[size]; szie -- ; down(1);`
4. 删除任意一个元素 — `heap[k] = heap[size]; size -- ; down(k); up(k);`
5. 修改任意一个元素 — `heap[k] = x; down(k); up(k);`

模拟堆

2024年2月22日 20:31

839. 模拟堆

题目 | 提交记录 | 讨论 | 题解

维护一个集合，初始时集合为空，支持如下几种操作：

1. “I x”，插入一个数x；
2. “PM”，输出当前集合中的最小值；
3. “DM”，删除当前集合中的最小值（当最小值不唯一时，删除最早插入的最小值）；
4. “D k”，删除第k个插入的数；
5. “C k x”，修改第k个插入的数，将其变为x；

现在要进行N次操作，对于所有第2个操作，输出当前集合的最小值。

输入格式

第一行包含整数N。

接下来N行，每行包含一个操作指令，操作指令为”I x”，”PM”，”DM”，”D k”或”C k x”中的一种。

输出格式

对于每个输出指令“PM”，输出一个结果，表示当前集合中的最小值。

每个结果占一行。

数据范围

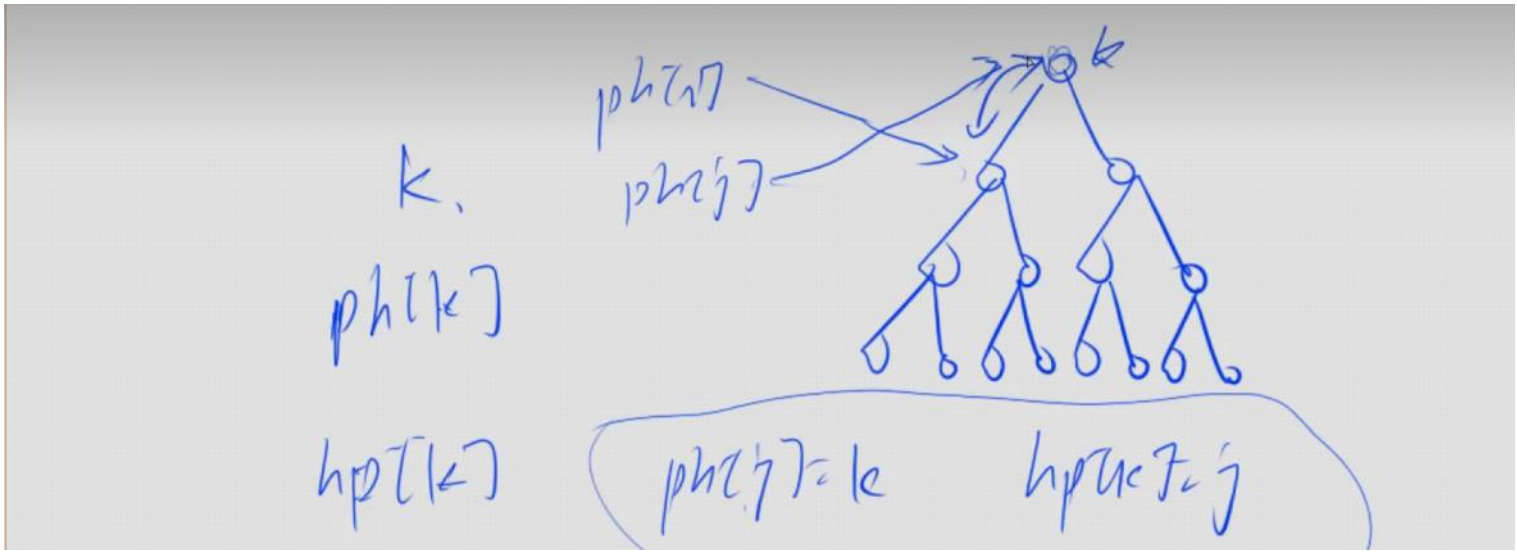
$1 \leq N \leq 10^5$
$-10^9 \leq x \leq 10^9$
数据保证合法。

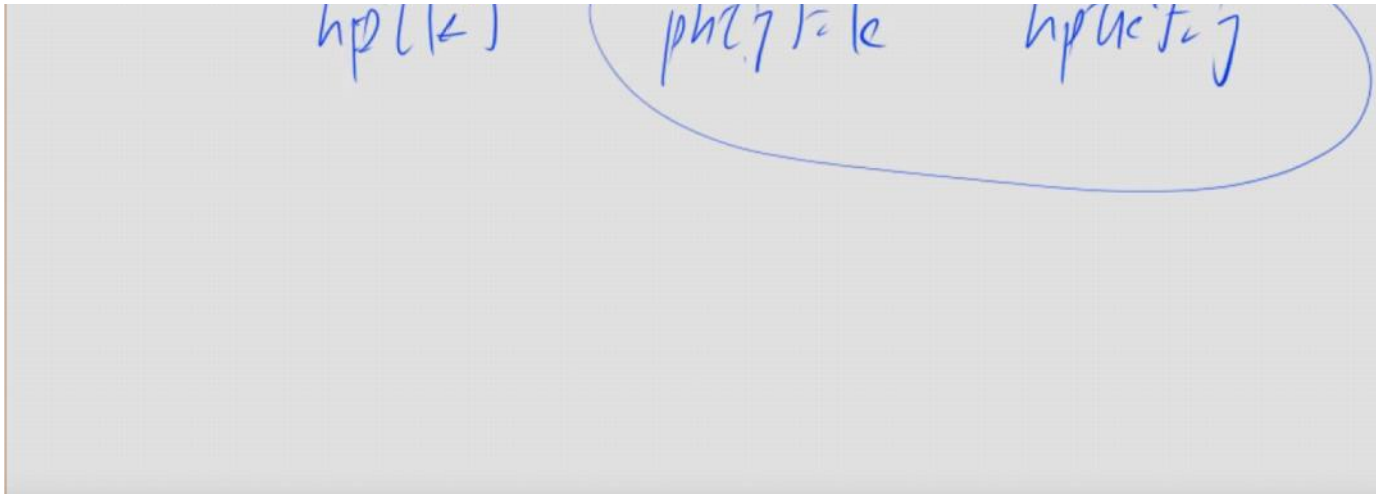
输入样例：

```
10
I -10
PM
I -10
D 1
C 2 8
I 6
PM
DM
```

输出样例：

```
-10
6
```





哈希

2024年2月22日 20:06

一般哈希

2024年2月22日 20:06

840. 模拟散列表

题目 | 提交记录 | 讨论 | 题解

维护一个集合，支持如下几种操作：

1. “I x”，插入一个数x；
2. “Q x”，询问数x是否在集合中出现过；

现在要进行N次操作，对于每个询问操作输出对应的结果。

输入格式

第一行包含整数N，表示操作数量。

接下来N行，每行包含一个操作指令，操作指令为”I x”，”Q x”中的一种。

输出格式

对于每个询问指令“Q x”，输出一个询问结果，如果x在集合中出现过，则输出“Yes”，否则输出“No”。

每个结果占一行。

数据范围

$1 \leq N \leq 10^5$
$-10^9 \leq x \leq 10^9$

输入样例：

```
5
I 1
I 2
I 3
Q 2
Q 5
```

输出样例：

```
Yes
No
```

10^5

$h(x) \in (0, 10^5)$

$-10^9 \sim 10^9$

① $x \bmod 10^5 \in [0, 10^5)$

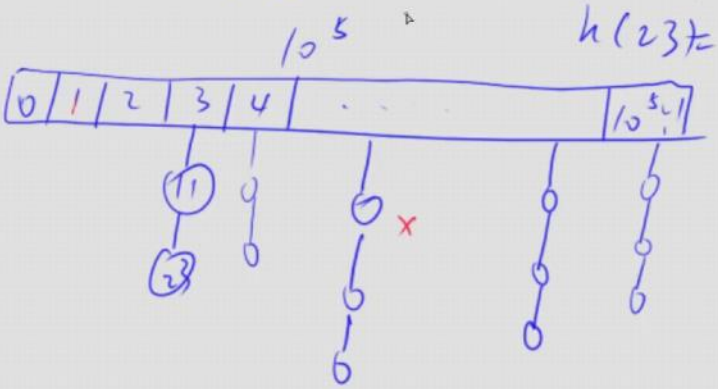
② 冲突

质数

$h(11) = 3$

$h(23) = 3$

拉链法



添加 $h(x)$

查找 $h(x)$

删除

字符串哈希

2024年2月22日 20:06

841. 字符串哈希

题目 | 提交记录 | 讨论 | 题解

给定一个长度为n的字符串，再给定m个询问，每个询问包含四个整数l_1, r_1, l_2, r_2，请你判断$[l_1, r_1]$和$[l_2, r_2]$这两个区间所包含的字符串子串是否完全相同。

字符串中只包含大小写英文字母和数字。

输入格式

第一行包含整数n和m，表示字符串长度和询问次数。

第二行包含一个长度为n的字符串，字符串中只包含大小写英文字母和数字。

接下来m行，每行包含四个整数l_1, r_1, l_2, r_2，表示一次询问所涉及的两个区间。

注意，字符串的位置从1开始编号。

输出格式

对于每个询问输出一个结果，如果两个字符串子串完全相同则输出“Yes”，否则输出“No”。

每个结果占一行。

数据范围

$1 \leq n, m \leq 10^5$

输入样例：

```
8 3
aabbaabb
1 3 5 7
1 3 6 8
1 2 1 2
```

输出样例：

```
Yes
No
Yes
```

Str = "ABCABCDEFXC Acwing"

h[0] = 0

h[1] = "A" 的 hash 值

h[2] = "AB" 的 hash 值

h[3] = "ABC"

h[4] = "ABCA"

⋮

"ABCD"

① P进制哈希

① 不能映射为0

A 0

AA 0

① A. B. C. D

(1 2 3 4)$_p$

0 ~ Q-1

$= (1\times p^3 + 2\times p^2 + 3\times p^1 + 4\times p^0) \bmod Q$.

Str = "ABCABCDEFXC Acwing"

h[0] = 0

h[1] = "A" 的 hash 值

h[2] = "AB" 的 hash 值

h[3] = "ABC"

h[4] = "ABCA"

⋮

"ABCD"

① P进制哈希

① 不能映射为0

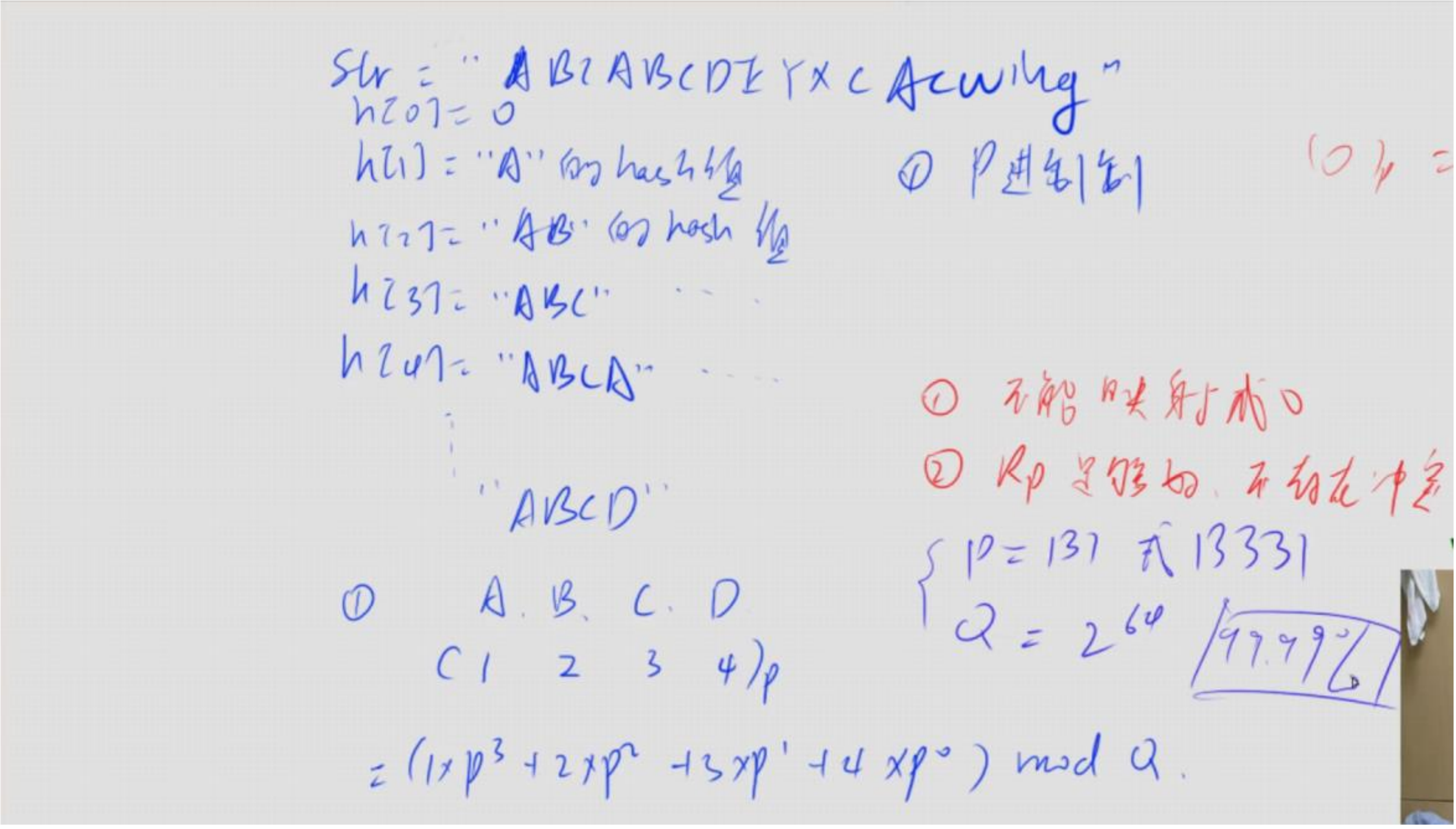
① A. B. C. D

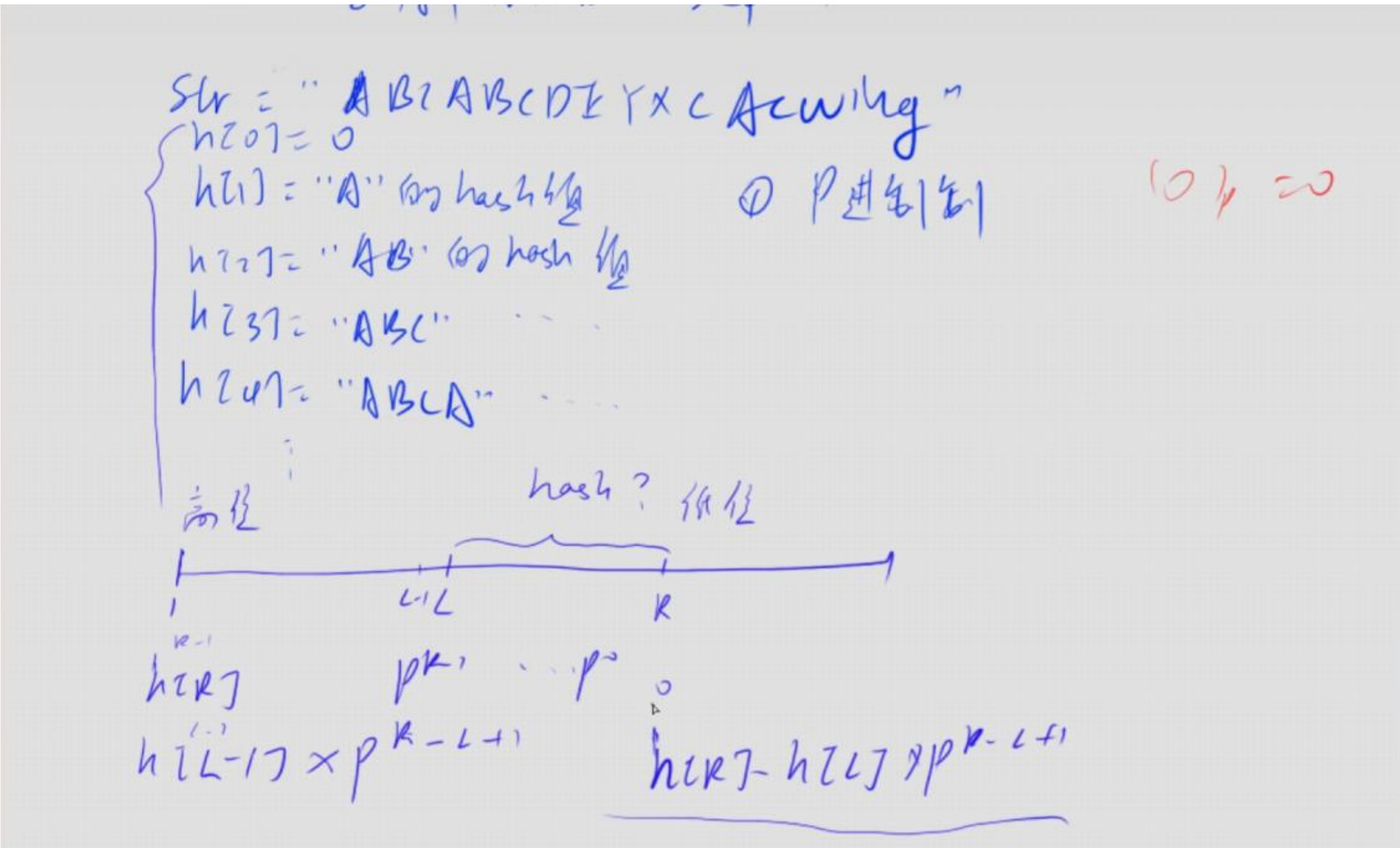
(1 2 3 4)$_p$

0 ~ Q-1

$= (1\times p^3 + 2\times p^2 + 3\times p^1 + 4\times p^0) \bmod Q$.

② Rp 冲突的概率很小


Str = "ABCABCDEYXCAcwing"
h[0] = 0
h[1] = "A" 的hash值
h[2] = "AB" 的hash值
h[3] = "ABC" ...
h[4] = "ABCA" ...
"ABCD"
① A B C D
(1 2 3 4)p
= (1×p^3 + 2×p^2 + 3×p^1 + 4×p^0) mod Q
① p进制制
① 不能映射成0
② Rp 足够好，不存在冲突
p = 131 或 13331
Q = 2^64
99.99%


Str = "ABCABCDEYXCAcwing"
h[0] = 0
h[1] = "A" 的hash值
h[2] = "AB" 的hash值
h[3] = "ABC" ...
h[4] = "ABCA" ...
① p进制制
高位
hash?
低位
1
L-1
L
R
h[R]
h[L-1] × p^(R-L+1)
h[R] - h[L] × p^(R-L+1)

STL简介

2024年2月22日 20:06

vector, 变长数组，倍增的思想

size() 返回元素个数

empty() 返回是否为空

clear() 清空

front()/back()

push_back()/pop_back()

begin()/end()

[]

支持比较运算，按字典序

pair<int, int>

first, 第一个元素

second, 第二个元素

支持比较运算，以first为第一关键字，以second为第二关键字（字典序）

string，字符串

size()/length() 返回字符串长度

empty()

clear()

substr(起始下标，(子串长度)) 返回子串

c_str() 返回字符串所在字符数组的起始地址

queue, 队列

size()
empty()
push() 向队尾插入一个元素
front() 返回队头元素
back() 返回队尾元素
pop() 弹出队头元素

priority_queue, 优先队列，默认是大根堆
push() 插入一个元素
top() 返回堆顶元素
pop() 弹出堆顶元素
定义成小根堆的方式：priority_queue<int, vector<int>, greater<int>> q;

stack, 栈
size()
empty()
push() 向栈顶插入一个元素
top() 返回栈顶元素
pop() 弹出栈顶元素

deque, 双端队列
size()
empty()
clear()
front()/back()
push_back()/pop_back()
push_front()/pop_front()

begin()/end()

[]

set, map, multiset, multimap, 基于平衡二叉树（红黑树），动态维护有序序列

size()

empty()

clear()

begin()/end()

++, -- 返回前驱和后继，时间复杂度 O(logn)

set/multiset

insert() 插入一个数

find() 查找一个数

count() 返回某一个数的个数

erase()

(1) 输入是一个数x，删除所有x O(k + logn)

(2) 输入一个迭代器，删除这个迭代器

lower_bound()/upper_bound()

lower_bound(x) 返回大于等于x的最小的数的迭代器

upper_bound(x) 返回大于x的最小的数的迭代器

map/multimap

insert() 插入的数是一个pair

erase() 输入的参数是pair或者迭代器

find()

[] 注意multimap不支持此操作。 时间复杂度是 O(logn)

lower_bound()/upper_bound()

unordered_set, unordered_map, unordered_multiset, unordered_multimap, 哈希表

和上面类似，增删改查的时间复杂度是 O(1)

不支持 lower_bound()/upper_bound()， 迭代器的++，--

bitset, 压位

bitset<10000> s;

~, &, |, ^

>>, <<

==, !=

[]

count() 返回有多少个1

any() 判断是否至少有一个1

none() 判断是否全为0

set() 把所有位置成1

set(k, v) 将第k位变成v

reset() 把所有位变成0

flip() 等价于~

flip(k) 把第k位取反