小土刀的面试刷题笔记 (index.html)

前期准备

解题思路

基础知识

数据库 (14520846750052.html)

iOS (14520846914394.html)

Objective-C (14520847557426.html)

Java (14520847149074.html)

数组和字符串

ZigZag Conversion (14520595481874.html)

Unique Character (14520595481622.html)

Text Justification (14520595481376.html)

!!Substring with Concatenation of All Words (14520595481118.html)

Submatrix Sum Zero (14520595480877.html)

Implement strStr() (14520595480652.html)

String Compression (14520595480445.html)

String Composition (14520595480211.html)

String Comparison (14520595479999.html)

Spiral Matrix (14520595479796.html)

Spiral Matrix II (14520595479570.html)

Simplify File Path (14520595479365.html)

Shortest Palindrome (14520595479135.html)

Set Matrix Zero (14520595478930.html)

01 相等的串 (14520595478736.html)

Rotate String (14520595478523.html)

Rotate Image (14520595478302.html)

Rotate Array (14520595478107.html)

Roman to Integer (14520595477901.html)

Reverse Sentence (14520595477695.html)

Reverse Integer (14520595477503.html)

Repeated DNA Sequences (14520595477318.html)

Reorder String by Case (14520595477117.html)

Reorder Rotated Array (14520595476933.html)

Remove Element (14520595476748.html)

Remove Duplicates from Sorted Array (14520595476547.html)

Remove Duplicates from Sorted Array II (14520595476340.html)

Peaks and Valleys (14520595476141.html)

Pascal's Triangle (14520595475941.html)

Pascal's Triangle II (14520595475767.html)

One Away (14520595475576.html)

奇偶调序 (14520595475403.html)

二进制矩阵中 1 的个数 (14520595475237.html)

Multiply Strings (14520595475077.html)

Move Zeroes (14520595474923.html)

Modify String (14520595474770.html)

Min Triangle Path Sum (14520595474607.html)

Min Difference 2 Array (14520595474431.html)

Merge Two Sorted Array (14520595474265.html)

Sorted Two Sorted Array II (14520595474103.html)

Merge Range (14520595473959.html)

Median of Unsorted Array (14520595473794.html)

最大间隔问题 (14520595473642.html)

Matrix ZigZag Traversal (14520595473495.html)

Majority Element (14520595473349.html)

Majority Element III (14520595473216.html)

Majority Element II (14520595473082.html)

Longest Word in Dictionary (14520595472951.html)

Longest Consecutive Sequence (14520595472823.html)

Longest Common Prefix (14520595472695.html)

Letter Combinations of a Phone Number (14520595472570.html)

Length of Last Word (14520595472449.html)

Largest Number (14520595472329.html)

K Sum II (14520595472209.html)

Isomorphic Strings (14520595472084.html)

Interleaving Array (14520595471966.html)

Integer to Roman (14520595471853.html)

Integer to English Words (14520595471740.html)

Insert Range (14520595471570.html)

Heapify (14520595471461.html)

H-Index (14520595471345.html)

H-Index II (14520595471233.html)

Gray Code (14520595471113.html)

Game of Life (14520595471005.html)

Find Anagram (14520595470900.html)

Excel Sheet Column Title (14520595470785.html)

Excel Sheet Column Number (14520595470670.html)

Evaluate Reverse Polish Notation (14520595470575.html)

Determine String Permutation (14520595470471.html)

Count and Say (14520595470378.html)

Compare Version Numbers (14520595470276.html)

Check Valid Sudoku (14520595470188.html)

Check Valid Number (14520595470088.html)

Check Permutation (14520595470004.html)

Palindrome Permutation (14520595469921.html)

Check Palindrome (14520595469833.html)

Palindrome Number (14520595469782.html)

Check Duplicate (14520595469735.html)

Find Duplicates (14520595469690.html)

Check Duplicate III (14520595469610.html)

Check Duplicate II (14520595469566.html)

Candy (14520595469517.html)

Binary to String (14520595469475.html)

Add One (14520595469437.html)

Add Binary (14520595469400.html)

4 Sum (14520595469361.html)

3 Sum (14520595469329.html)

3 Sum Closest (14520595469298.html)

2 Sum (14520595469247.html)

堆栈和队列

Implement Stack using Queues (14520606685840.html)

Implement Stack using Queues (14520606685779.html)

Stack of Plates (14520606685720.html)

Sort Stack (14520606685665.html)

Queue of Stack (14520606685606.html)

Min Stack (14520606685544.html)

Max Stack (14520606685475.html)

Longest Valid Parentheses (14520606685419.html)

In Order Travesal with Stack (14520606685367.html)

Hanoi Tower with Stack (14520606685319.html)

Basic Calculator (14520606685269.html)

Basic Calculator II (14520606685216.html)

链表

Swap Adjacent Node (14520597852720.html)

Start of Circle (14520597852604.html)

Sort List (14520597852502.html)

Rotate List (14520597852382.html)

Reversely List Traverse (14520597852276.html)

Reverse Nodes in k-Group (14520597852177.html)

Reverse List (14520597852065.html)

Reverse List Range (14520597851968.html)

Reorder List (14520597851880.html)

Remove Linked List Elements (14520597851782.html)

Remove Duplicates from Unsorted List (14520597851695.html)

Remove Duplicates from Sorted List (14520597851601.html)

Remove Duplicates from Sorted List II (14520597851549.html)

Partition Linked List (14520597851500.html)

Partitiom List Sorted (14520597851449.html)

Middle of List (14520597851399.html)

Merge Two Lists (14520597851346.html)

Merge K Linked List (14520597851075.html)

Kth to Last (14520597851031.html)

Insertion Sort List (14520597850990.html)

Copy List with Random Pointer (14520597850946.html)

Palindrome Linked List (14520597850907.html)

Check Intersection (14520597850871.html)

Check Cycle (14520597850832.html)

Add Two Numbers (14520597850789.html)

递归和动态规划

Word Break (14520604921847.html)

Word Break II (14520604921620.html)

Wildcard Matching (14520604921399.html)

Unique Path (14520604921164.html)

Unique Path II (14520604920947.html)

Ugly Number (14520604920730.html)

Ugly Number II (14520604920507.html)

Trap Water (14520604920305.html)

Towers of Hanoi (14520604920091.html)

Surrounded Regions (14520604919872.html)

Continuous Subarray Sum (14520604919653.html)

Subarray Sum to K (14520604919436.html)

Stack of Box (14520604919240.html)

Restore IP Addresses (14520604919038.html)

Regular Expression Matching (14520604918826.html)

Recursive Multiply (14520604918618.html)

Product of Array Except Self (14520604918412.html)

Pow(x, n) (14520604918229.html)

Palindrome Partitioning II (14520604918057.html)

Palindrom Partition (14520604917885.html)

Paint Fill (14520604917714.html)

Package Problem (14520604917521.html)

Package Problem II (14520604917350.html)

Number of Islande (14520604917181.html)

Nth Prime (14520604917011.html)

N Queen (14520604916850.html)

N-Queens II (14520604916662.html)

Minimum Subarray for Sum (14520604916479.html)

Minimum Window Substring (14520604916322.html)

Min Sum Subarray (14520604916164.html)

Min Adjustment Cost (14520604916010.html)

Maximum Subarray (14520604915856.html)

Max Sum Subarray Index (14520604915702.html)

Max Sum 2 Subarray (14520604915546.html)

Max Square (14520604915394.html)

Maximal Rectangle (14520604915232.html)

Max Product Subarray (14520604915082.html)

Max Difference 2 Subarray (14520604914933.html)

Longest Substring Without Repeating Characters (14520604914788.html)

Longest Substring with K Unique Characters (14520604914642.html)

Longest Palindromic Substring (14520604914478.html)

Longest Increasing Sequence (14520604914337.html)

Longest Increasing Consecutive Sequence (14520604914202.html)

Longest Common Substring (14520604914067.html)

Longest Common Subsequence (14520604913939.html)

Largest Rectangle in Histogram (14520604913809.html)

Kth Permuation (14520604913655.html)

Jump Game (14520604913528.html)

Jump Game II (14520604913406.html)

Interleaving String (14520604913280.html)

House Robbery (14520604913150.html)

House Robbery II (14520604912979.html)

Generate Parentheses (14520604912864.html)

Gas Station (14520604912743.html)

Fibonacci (14520604912625.html)

Expression Add Operators (14520604912510.html)

Edit Distance (14520604912393.html)

Dungeon Game (14520604912257.html)

Different Ways to Add Parentheses (14520604912153.html)

Different Subsequence (14520604912043.html)

Decode Ways (14520604911938.html)

Container with Most Water (14520604911834.html)

Combinations (14520604911732.html)

Combination Sum (14520604911617.html)

Combination Sum III (14520604911516.html)

Combination Sum II (14520604911420.html)

Coin (14520604911320.html)

Coin Game (14520604911261.html)

Coin Game II (14520604911204.html)

Climb Stairs (14520604911145.html)

Climb Stairs - Triple Step (14520604911084.html)

Boggle Game - Word Search (14520604911019.html)

Boggle Game: Word Search II (14520604910956.html)

Best Time to Buy and Sell Stock (14520604910906.html)

Best Time to Buy and Sell Stock IV (14520604910834.html)

Best Time to Buy and Sell Stock III (14520604910785.html)

Best Time to Buy and Sell Stock II (14520604910739.html)

All Subsets (14520604910695.html)

All Permutation (14520604910652.html)

All Permutations II (14520604910607.html)

排序和搜索

URL Duplicates (14520606004836.html)

Stream Integer (14520606004678.html)

Sqrt(x) (14520606004545.html)

Sparse Search (14520606004412.html)

Sorted Search, No Size (14520606004279.html)

Sort Colors (14520606004147.html)

Sort Colors II (14520606003957.html)

Sort Big File (14520606003841.html)

Sort Age (14520606003704.html)

Social Network (14520606003582.html)

Search Rotated Array (14520606003458.html)

Search Range (14520606003336.html)

Search Insert Position (14520606003219.html)

Search 2D matrix (14520606003106.html)

Recursive Integer Traversal (14520606002989.html)

Nuts & Bolts Problem (14520606002879.html)

Missing Number (14520606002777.html)

Find Minimum in Rotated Sorted Array (14520606002660.html)

Median of Two Sorted Array (14520606002549.html)

Maximum Gap (14520606002445.html)

Kth Smallest Number (14520606002338.html)

Kth Largest in Sorted Matrix (14520606002263.html)

Kth Largest Element (14520606002197.html)

Inverted Index (14520606002136.html)

Index Equals to Value (14520606002077.html)

Hit Counter (14520606002015.html)

Group Anagrams (14520606001958.html)

Group Anagrams List (14520606001885.html)

First Missing Positive Number (14520606001824.html)

First Error Version (14520606001769.html)

Find Peak Element (14520606001720.html)

名人问题 (14520606001667.html)

Count Plane (14520606001623.html)

Binary Search (14520606001575.html)

树和图

Zigzag Level Order Traversal (14520607221934.html)

Word Ladder (14520607221747.html)

Word Ladder II (14520607221562.html)

Valid Binary Search Tree (14520607221359.html)

Unique Binary Search Trees (14520607221154.html)

Unique Binary Search Trees II (14520607220949.html)

Implement Trie (Prefix Tree) (14520607220765.html)

Topological Sort (14520607220590.html)

Convert Sorted List to Binary Search Tree (14520607220397.html)

Sorted Array to Binary Search Tree (14520607220201.html)

Serialize and Deserialize (14520607220001.html)

Search Range (14520607219827.html)

Scramble String (14520607219664.html)

Route Between Nodes (14520607219495.html)

Root Path Sum (14520607219333.html)

Binary Tree Right Side View (14520607219173.html)

Reverse Binary Tree (14520607219009.html)

Recover Binary Search Tree (14520607218845.html)

Preorder Traversal (14520607218687.html)

Postorder Traversal (14520607218530.html)

Node Path Sum (14520607218372.html)

Paths with Sum II (14520607218218.html)

Populating Next Right Pointers in Each Node (14520607218066.html)

Next Node without Parent (14520607217915.html)

Next Node (14520607217765.html)

Next Node in BST (14520607217620.html)

Neighbor Node (14520607217464.html)

Minimum Path Sum (14520607217318.html)

Min Height of Binary Tree (14520607217172.html)

Maximum Path Sum (14520607217030.html)

Height of Binary Tree (14520607216892.html)

Level Order Traversal (14520607216756.html)

Level Order Traversal II (14520607216621.html)

Leaf Path Sum (14520607216488.html)

Kth Smallest Element in a BST (14520607216345.html)

Is Subtree (14520607216180.html)

Insert node to BST (14520607216053.html)

Inorder Traversal (14520607215936.html)

Flatten Binary Tree to Linked List (14520607215811.html)

First Common Ancestor (14520607215690.html)

First Common Ancestor of a Binary Search Tree (14520607215576.html)

Different BST (14520607215459.html)

Different BST II (14520607215341.html)

Delete Node in a Linked List (14520607215217.html)

Delete Middle Node (14520607215107.html)

Cut Wood (14520607214993.html)

Course Schedule (14520607214882.html)

Course Schedule II (14520607214780.html)

Connected Nodes in Undirected Graph (14520607214651.html)

Complete Tree Node Count (14520607214579.html)

Clone Graph (14520607214515.html)

Symmetric Tree (14520607214451.html)

Same Tree (14520607214387.html)

Build Order (14520607214326.html)

BST Sequences (14520607214254.html)

Binary Tree to Linked List (14520607214194.html)

Construct Binary Tree from Preorder and Inorder Traversal (14520607214136.html)

Construct Binary Tree from Inorder and Postorder Traversal (14520607214067.html)

Binary Tree Depth (14520607214003.html)

Binary Search Tree Iterator (14520607213955.html)

Balance Tree (14520607213899.html)

All Path (14520607213844.html)

位操作

Swap Bits (14520596469127.html)

Square of Two (14520596469033.html)

Single Element (14520596468981.html)

Single Element III (14520596468931.html)

Single Element II (14520596468879.html)

Set Bits (14520596468825.html)

Reverse Bits (14520596468781.html)

Number of One (14520596468734.html)

Next Number (14520596468686.html)

Flip Bits (14520596468613.html)

Draw Line (14520596468570.html)

Divide Two Integers (14520596468530.html)

Check Power of Two (14520596468493.html)

Bitwise AND of Numbers Range (14520596468456.html)

A Plus B (14520596468418.html)

数学逻辑

Remove Digit (14520603231966.html)

Quick Pow (14520603231831.html)

Previous Permuation (14520603231697.html)

Poison (14520603231576.html)

Permutation Sequence (14520603231466.html)

Perfect Squares (14520603231334.html)

Number of Digit (14520603231230.html)

Number of Digit One (14520603231129.html)

Next Permutation (14520603231017.html)

Max Points on a Line (14520603230913.html)

Jugs of Water (14520603230810.html)

Index of Permuation (14520603230712.html)

Index of Permutation II (14520603230618.html)

The Heavy Pill (14520603230526.html)

Hash Function (14520603230426.html)

Happy Number (14520603230367.html)

Fraction to Recurring Decimal (14520603230311.html)

Factorial Trailing Zeroes (14520603230254.html)

The Egg Drop Problem (14520603230202.html)

Divide Number (14520603230148.html)

均分 01 (14520603230094.html)

Count Primes (14520603230019.html)

Cosine Similarity (14520603229974.html)

Basketball (14520603229929.html)

The Apocalypse (14520603229887.html)

Ants on a Triangle (14520603229844.html)

Add Digits (14520603229804.html)

100 Lockers (14520603229761.html)

面向对象设计

Web Crawler (14520604447774.html)

Tiny URL (14520604447653.html)

Stock Data (14520604447530.html)

Social Network (14520604447397.html)

Singleton (14520604447289.html)

Sales Rank (14520604447186.html)

Rank from Stream (14520604447071.html)

Personal Financial Manager (14520604446948.html)

Peeking Iterator (14520604446839.html)

Pastebin (14520604446725.html)

Parking Lot (14520604446608.html)

Othello (14520604446509.html)

Online Book Reader (14520604446403.html)

Music Player (14520604446344.html)

Minesweeper (14520604446282.html)

LRU Cache (14520604446223.html)

Jukebox (14520604446169.html)

Jigsaw (14520604446101.html)

Hash Table (14520604446041.html)

File System (14520604445970.html)

Elevator (14520604445920.html)

Deck of Cards (14520604445872.html)

Chat Server (14520604445829.html)

Call Center (14520604445788.html)

Animal Shelter (14520604445745.html)

Add and Search Word - Data structure design (14520604445699.html)

测试

Web Browser (14520607787281.html)

Pen (14520607787230.html)

Login System (14520607787180.html)

Binary Search (14520607787127.html)

网络

TCP or UDP (14520603814674.html)

Reliable UDP (14520603814632.html)

After URL (14520603814585.html)

公司介绍及面经

Zillow (14520850403309.html)

Zazzle (14520850403158.html)

Yelp (14520850403014.html)

Yahoo (14520850402866.html)

Xcalar (14520850402716.html)

White Pages (14520850402570.html)

微信 (14520850402406.html)

Visual Concepts (14520850402230.html)

Uber (14520850402072.html)

Twitter (14520850401647.html)

Raventech 渡鸦科技 (14520850401521.html)

PocketGem (14520850401393.html)

Oracle (14520850401269.html)

Microsoft (14520850401149.html)

Liveramp 面经 (14520850401028.html)

Linkedin (14520850400912.html)

汇丰银行 (14520850400786.html)

Google (14520850400653.html)

Facebook (14520850400582.html)

Expensify (14520850400516.html)

Exablox (14520850400441.html)

Epic (14520850400373.html)

Dropbox (14520850400303.html)

大疆 (14520850400236.html)

cPacket (14520850400159.html)

Caliper Corporation (14520850400097.html)

BrightEdge (14520850400034.html)

Bloomberg (14520850399984.html)

Amazon (14520850399925.html)

Amazon OA (14520850399861.html)

数组和字符串

解题策略

- 一般来说，一旦出现“unique”，就落入使用哈希表或者bitset来判断元素出现与否的范畴。
- 一旦需要统计一个元素集中元素出现的次数，我们就应该想到哈希表。

数组

在C／C++中，标准的数组可以通过在栈(Stack)上分配空间，或者通过先声明指针，然后用new关键字(或者C中的malloc函数)，在堆(Heap)上动态的分配空间。举例如下：

```
// 在栈上定义长度为arraySize的整型数组
int array[arraySize];
// 在堆上定义长度为arraySize的整型数组
int *array = new int[arraySize];
```

使用完后需要释放内存：

```
delete[] array;
```

访问数组时要注意越界问题。

数组可以通过下标随机访问元素，所以在修改、读取某个元素的时候效率很高，具有O(1)的时间复杂度。在插入、删除的时候需要移动后面的元素，所以平均时间复杂度O(n)。通常，数组这个数据结构不是很适合增减元素。如果你想要的操作需要大量在随机位置增减元素，可以考虑其他的数据结构。在C++中，标准库提供vector容器，其本质上相当于一个长度可变的动态数组。

哈希表

~~哈希表(Hash Table)几乎是最为重要的数据结构，主要用于基于“键(key)”的查找，存储的基本元素是键-值对(key-value pair)。逻辑上，数组可以作为哈希表 的一个特例：键是一个非负整数。注意，通常哈希表会假设键是数据的唯一标识，相同的键默认表示同一个基本存储元素。~~

~~哈希表的本质是当使用者提供一个键，根据哈希表自身定义的哈希函数(hash function)，映射出一个下标，根据这个下标决定需要把当前的元素存储在什么位置。在一些合理的假设情况下，查~~找一个元素的平均时间复杂度是O(1)，插入一个元素的平摊(amortized)时间复杂度是O(1)。

当对于不同的键，哈希函数提供相同的存储地址时，哈希表就遇到了所谓的冲突(collision)。解决冲突的方式有链接法(chaining)和开放地址法(open addressing)两种。简单来说，链接法相当于利用辅助数据结构(比如链表)，将哈希函数映射出相同地址的那些元素链接起来。而开放地址法是指以某种持续的哈希方式继续哈希，直到产生的下标对应尚未被使用的存储地址，然后把当前元素存储在这个地址里。

通常而言，链接法实现相对简便，但是可能需要附加空间，并且利用当前空间的效率不如开放地址法高。开放地址法更需要合理设计的连续哈希函数，但是可以获得更好的空间使用效率。需要注意的是，过于频繁的冲突会降低哈希表的搜索效率，此时需要哈希表的扩张。

C++标准库中提供map容器，可以插入、删除、查找键-值对，底层以平衡二叉搜索树的方式实现，根据键进行了排序。严格来说，map并不是一个哈希表，原因是查找时间从O(1)变为了O(log n)，但是好处在于能够根据键值，顺序地输出元素，对于某些应用场景可能更为合适。在C++11中，标准库添加了unordered_map，更符合哈希表的传统定义，平均查找时间O(1)。

字符串

在C语言中，字符串(string)指的是一个以‘\0’结尾的char数组。关于字符串的函数通常需要传入一个字符型指针。然而，在C++中，String是一个类，并且可以通过调用类函数实现判断字符串长度，字串等等操作。

工具箱

栈和堆

栈主要是指由操作系统自动管理的内存空间。当进入一个函数，操作系统会为该函数中的局部变量分配存储空间。事实上，系统会分配一个内存块，叠加在当前的栈上，并且利用指针指向前一个内存块的地址。函数的局部变量就存储在当前的内存块上。当该函数返回时，系统“弹出”内存块，并且根据指针回到前一个内存块。所以，栈总是以后进先出(LIFO)的方式工作。通常，在栈上分配的空间不需要用户操心。

堆是用来存储动态分配变量的空间。对于堆而言，并没有像栈那样后进先出的规则，程序员可以选择随时分配或回收内存。这就意味着需要程序员自己用命令回收内存，否则会产生内存泄漏(memory leak)。在C/C++中，程序员需要调用free/delete来释放动态分配的内存。在Java, Objective-C (with Automatic Reference Count)中，语言本身引入垃圾回收和计数规则帮助用户决定在什么时候自动释放内存。

需要了解的常见容器及方法

- Vector / ArrayList
- HashSet, HashTable
- Map / Dictionary

C 字符串常见函数

“相关函数通常定义在string.h中，简要介绍如下常见函数，更多信息请参考这里(http://www.cplusplus.com/reference/cstring/)：

```
//copy source string to destination string
char *strcpy ( char *destination, const char *source );

Example:
char str1[] = "Sample string";
char str2[SAMPLE_STRING_SIZE];
strcpy (str2,str1);

//Appends a copy of the source string to the destination string.
char *strcat ( char *destination, const char *source );

Example:
char str[STRING_SIZE] = "string is ";
strcat(str, "concatinated");    //str becomes "string is concatinated

// Returns a pointer to the first occurrence of character in the C string str (N
char *strchr ( const char *str, int character );

Example:
char string[STRING_LENGTH] = "This is a string";
int firstOccurPos = (int)(strchr(string, 'i') - string);    // firstOccurPos equ

// Returns a pointer to the last occurrence of character in the C string str.
char *strrchr ( char *str, int character );

// Returns a pointer to the first occurrence of str2 in str1, or a NULL pointer
char *strstr (char *str1, const char *str2 );

// Returns the length of the C string str.
size_t strlen ( const char *str );

Example:
char string[STRING_LENGTH] = "This is a string";
int length = strlen(string);    // length equals to 16

// convert char string to a double
double atof (const char *str);

// convert char string to an int
int atoi (const char *str);
```

C++ String 类常见函数

由于String类重载了+, <, >, =, ==等运算符，故复制，比较是否相等，附加子串等都可以用运算符直接实现。简要介绍其他常见函数如下，更多信息请参考这里 (http://www.cplusplus.com/reference/string/string/)：

```
// Searches the string for the first occurrence of the str, returns index
size_t find (const string& str, size_t pos = 0) const;

// Returns a newly constructed string object with its value initialized to a cop
string substr (size_t pos = 0, size_t len = npos) const;

Example:
string str("This is a string");
string subStr = str.substr(10,6);    // subStr equals to "string", with length 6

// erase characters from pos with length len
string &erase (size_t pos = 0, size_t len = npos);

Example:
string str("This is a string");
str.erase(0,10);    // str becomes "string", with length 6 after erasure

// Returns the length of the string, in terms of bytes.
size_t length();

Example:
string str ("This is a string");
int length = str.length();    // length equals to 16
```

String Matching 的常见算法

简单介绍两种比较易于实现的字符串比较算法，下述算法假设在长度为n的母串中匹配长度为m的子串。更详细的解释请见这里 (http://en.wikipedia.org/wiki/String_searching_algorithm)。

Brute-Force算法： 顺序遍历母串，将每个字符作为匹配的起始字符，判断是否匹配子串。时间复杂度 O(mn)。

Rabin-Karp算法：将每一个匹配子串映射为一个哈希值。例如，将子串看做一个多进制数，比较它的值与母串中相同长度子串的哈希值，如果相同，再细致地按字符确认字符串是否确实相同。顺序计算母串哈希值的过程中，使用增量计算的方法：扣除最高位的哈希值，增加最低位的哈希值。因此能在平均情况下做到O(m+n)。

Example:

为描述简单，假设字符串只含有十进制数字，母串为“1234”待匹配子串为“23”，定义hash函数把字符串转成整数值。

首先计算子串hash：值为23。

然后计算母串前两个字符的hash：值为12，与子串不符合，需要后移。此时我们扣除最高位“1”的“hash，增加新的最低位“3”的hash，得到新的hash值23，与子串相符。

通过按字符比较发现的确匹配成功，可以返回母串匹配上的下标。

伪代码：

```
int RabinKarp(string s[1..n], string pattern[1..m])
hpattern = hash(pattern[1..m]);  hs = hash(s[1..m]);
for i from 1 to n−m+1
if hs = hpattern
     if s[i..i+m−1] = pattern[1..m]
return i
hs = hash(s[i+1..i+m])
       return not found
```

« Binary Search (14520607787127.html)

Zigzag Level Order Traversal » (14520607221934.html)

TOP