**Onsite**  
**First set**  
First round:  
Reverse Polish Notation  
  
**Make sure to know what the requirement is.**  
  
Operator (base class)  
- operate(int val1, int val2)  
  
Add,Subtract, Multiply, Divide四个class继承这个Operator base class的写法  
  
- pay attention to Divide  
. visit [1point3acres.com](http://1point3acres.com/) for more.  
**Don’t over-design!**  
  
follow ups:  
- token, operand and operator inherit from token  
- unary and ternary operators, add a variable of NumOperands  
- factory design pattern

Factory design pattern is a class where you can hide the creation logic of all sub-classes. Typically it would require a type and generate an object of sub-class. Essentially it’s a mapping from types sub-classes.

- 怎么才能直接给用户binary, 让他们可以自由的添加新的operator，更好的办法是做XML / JSON的serialization  
  
-google 1point3acres  
**Take a look at how to use Abstract class.**  
. 涓€浜�-涓夊垎-鍦帮紝鐙鍙戝竷  
design题：  
一个Calculator类，包含一个stack和一个vector<Token>。. from: [1point3acres.com/bbs](http://1point3acres.com/bbs)   
  
一个 Token 类，包含一个process(stack)方法。  
Operand 和 Operator 继承自Token。Operand的process是向stack中push这个数字。Operator包含一个numOfOperand，process方法是从stack中pop出numOfOperand个数字后进行某种操作，结果再push进stack。  
  
Remove subtree from tree  
  
  
给的程序是用C写的，很长，大部分不用看，只需要写一个子函数，要现场编译现场跑给定一个数组，数组的每个元素就是一个节点(struct node)，大概的长得像下面这样  
  
struct node{  
    int parent;  
    int val;  
    bool valid;  
};. From 1point 3acres bbs  
  
parent代表当前node的parent 在数组里的index，root node的parent是-1. 所以node 是child指向parent的。给定一个数组和数组的某个index，删除这个index以及它的子树(只需要将node里的valid置为false即可)，只能用O(n)的空间复杂度  
  
解法：在strcut node里面添加一个新的元素visited（一定要记得在程序里初始化node的地方，把visited设为false），代表该node是否被访问过。然后从头到尾访问输入数组。对于当前访问的元素，如果已经被visited了，则忽略。否则，沿着parent指针走，直到到达根节点（则从当前node到根节点的所有node都不需要delete），或者到达一个被标记为删除的节点（则当前node到根节点所有的node都需要被删除）  
.鐣欏璁哄潧-涓€浜�-涓夊垎鍦�  
struct node  
{  
    int parent;. 涓€浜�-涓夊垎-鍦帮紝鐙鍙戝竷  
    int val;  
    bool valid;. From 1point 3acres bbs  
    bool visited;  
};  
  
要实现的函数大概长这样：  
鏉ユ簮涓€浜�.涓夊垎鍦拌鍧�.   
void DeleteSubTree(node\* head, int index, int n){  
    // n代表数组里面一共有n个元素  
    (head+index)->visited = true;   // Set up  
    (head+index)->valid = false;    // this first!!  
  
    for(int i=0; i < n; i++){  
        if ((head+i)->visited) continue; // Memorization  
  
        if(NeedDelete(head, i)){. 鐣欏鐢宠璁哄潧-涓€浜╀笁鍒嗗湴  
            label(head, i);  
        }.鐣欏璁哄潧-涓€浜�-涓夊垎鍦�  
    }  
}  
. visit [1point3acres.com](http://1point3acres.com/) for more.  
bool NeedDelete(node\* head, int index){. from: [1point3acres.com/bbs](http://1point3acres.com/bbs)   
    while(index != -1 && !(head+index)->visited ){ 鏉ユ簮涓€浜�.涓夊垎鍦拌鍧�.   
        (head+index)->visited = true;  
        index = (head+index)->parent;  
    }  
  
    if(index == -1) return false;  // If finds root, return False  
    return !(head+index)->valid;  
}  
  
void label(node\* head, int index){  
    while((head+index)->valid){  // Assumption!  
        (head+index)->valid = false;  
        index = (head+index)->parent;  
    }  
}. 涓€浜�-涓夊垎-鍦帮紝鐙鍙戝竷  
鏉ユ簮涓€浜�.涓夊垎鍦拌鍧�.   
. 涓€浜�-涓夊垎-鍦帮紝鐙鍙戝竷  
  
Follow ups:  
- update tree size (capacity)  
- corner case:

1. index不合法
2. 删除一个已经被删除的subtree的时候size会继续往下减

. from: [1point3acres.com/bbs](http://1point3acres.com/bbs)   
delete sub tree  
改写代码题：不能在node里加入需要动态分配内存的数据结构。用左儿子右兄弟解决。.鐣欏璁哄潧-涓€浜�-涓夊垎鍦�  
  
struct node {  
        value: int  
        isValid: bool  
        leftChild, rightSibling: node\*  
        upParent: node\*  
}  
  
linkNodes(child, parent) {. 鍥磋鎴戜滑@1point 3 acres  
        child.upParent = parent  
        child.rightSibling = parent.leftChild  
        parent.leftChild = child  
}. From 1point 3acres bbs  
  
removeNode(node) {.1point3acres缃�  
        node.isValid = false  
        node = node.leftChild  
        while (node != NULL) {. 1point 3acres 璁哄潧  
                remove(node)  
                node = node.rightSibling  
        }  
  
}  
  
  
Second round  
Two queues get elements with distance <= 1  
  
**Assuming we already get all elements on hand.**  
  
cur\_ts = blocking\_queue1.getNext()  
q1.append(cur\_ts).1point3acres缃�  
  
while q2 and cur\_ts - q2[0] > 1: # pop() for better performance in the future, explain it!!  
    q2.popleft() # deque, for performance  
  
  
for ts in q2:  
    if abs(ts - cur\_ts) <= 1:  
  
        print ts, cur\_ts. visit [1point3acres.com](http://1point3acres.com/) for more.  
  
    else:  
. more info on [1point3acres.com](http://1point3acres.com/)  
        break  
  
  
**Now, two blocking queues!**  
  
We use two threads, to avoid being blocked.  
  
def retrieve\_from\_queue1(stream):. 涓€浜�-涓夊垎-鍦帮紝鐙鍙戝竷  
    ts = stream.getNext() 鏉ユ簮涓€浜�.涓夊垎鍦拌鍧�.   
    calculate\_pairs(q2, q1, ts)  
  
Same thing for queue2.  
  
Meanwhile, two threads are modifying the same data structure — q1 and q2, so we need lock.. more info on [1point3acres.com](http://1point3acres.com/)

- lock = threading.Lock()  
- modify calculate\_pair to “with lock:”  
- finally start the threads, and run

try:

    thread.start\_new\_thread(thread1, (s1, ))

    thread.start\_new\_thread(thread2, (s2, ))

except:

    print "Couldn't start threads."

**各种切入点啊，看题目要求吧！**  
  
Followups:  
- 如果有多个queue，比如10个queue怎么办，lock ds (like semaphore), then do the same thing  
  
  
  
process() {  
        while (true) {  
                LOCK L1  
                x = s1.get()  
                list1.add(x)  
                UNLOCK L1. 鍥磋鎴戜滑@1point 3 acres  
  
                LOCK L2  
                for y in list2  
                        if (abs(x - y) < 1)  
                                print  
                UNLOCK L2  
        }  
  
}  
. Waral 鍗氬鏈夋洿澶氭枃绔�,  
  
Slow Web Accessing  
  
  
- client-server-database model  
- draw diagram/model  
. 1point 3acres 璁哄潧

. 鐗涗汉浜戦泦,涓€浜╀笁鍒嗗湴

(Start from browser)  
  
DNS (back to front)

- cache  
- test: check response time

CDN (ip address of a data center)

- cache  
- test: try different ip of the host

Load balancer / Router (inside a data center)

- different kinds of request

- download, long connection, web browsing, short connection  
- different protocol

- balance request numbers to different servers. more info on [1point3acres.com](http://1point3acres.com/)  
- test: monitor server load and request type

Back end

- asynchronous vs synchronous response  
- reduce I/O waiting time

Message queue

- distributed system

Database

- same request -> cache  
- too many connections at the same time -> batch  
- Database itself is slow -> in-memory database  
- bad formatted, low efficient queries  
  
- bad table design

(Back to browser)  
  
  
Front end

- bad javascript  
- call library, which doesn’t do well in CDN

Browser

- hardware acceleration

-google 1point3acres  
General / QoS

- bandwidth, low speed for transferring data  
- throughput, low speed for processing data-google 1point3acres  
(- error rate)

-google 1point3acres  
  
  
Wildcard matching  
  
**Firstly, think this through. All corner cases, test cases.**  
  
  
    def test\_empty\_string\_match(self):  
        self.assertTrue(is\_match('', ''))  
        self.assertFalse(is\_match('', '?'))  
        self.assertTrue(is\_match('', '\*'))  
        self.assertFalse(is\_match('', 'a\*'))  
        self.assertTrue(is\_match('', '\*\*'))  
. 涓€浜�-涓夊垎-鍦帮紝鐙鍙戝竷  
    def test\_exact\_match(self):. 鐣欏鐢宠璁哄潧-涓€浜╀笁鍒嗗湴  
        self.assertTrue(is\_match('a', 'a'))  
        self.assertTrue(is\_match('abc', 'abc'))-google 1point3acres  
        self.assertFalse(is\_match('abc', 'abb'))  
        self.assertFalse(is\_match('a', ''))  
  
    def test\_question\_mark\_match(self):  
        self.assertTrue(is\_match('a', '?'))  
        self.assertFalse(is\_match('ac', '?'))  
        self.assertFalse(is\_match('a', 'a?'))  
        self.assertFalse(is\_match('a', '?a'))  
        self.assertFalse(is\_match('a', '??'))  
        self.assertTrue(is\_match('ab', '??'))  
. From 1point 3acres bbs  
    def test\_star\_match(self):  
        self.assertTrue(is\_match('', '\*'))  
        self.assertTrue(is\_match('a', '\*'))  
        self.assertTrue(is\_match('abc', '\*')). more info on [1point3acres.com](http://1point3acres.com/)  
        self.assertTrue(is\_match('abc', 'a\*'))  
        self.assertFalse(is\_match('abc', 'b\*'))  
        self.assertTrue(is\_match('abc', '\*c'))  
        self.assertFalse(is\_match('abc', '\*d')). [1point3acres.com/bbs](http://1point3acres.com/bbs)  
        self.assertTrue(is\_match('abbbdddccc', 'a\*c'))  
        self.assertFalse(is\_match('abc', 'a\*d')). From 1point 3acres bbs  
  
    def test\_question\_mark\_and\_star(self):. 鐣欏鐢宠璁哄潧-涓€浜╀笁鍒嗗湴  
        self.assertTrue(is\_match('a', '?\*'))  
        self.assertTrue(is\_match('a', '\*?'))  
        self.assertTrue(is\_match('ab', '\*?'))  
        self.assertFalse(is\_match('a', '??\*'))  
        self.assertTrue(is\_match('abc', 'a?\*c'))  
        self.assertFalse(is\_match('abc', 'a?\*d'))  
        self.assertFalse(is\_match('abc', '?\*?d'))  
        self.assertTrue(is\_match('abc', 'a\*\*\*?c’))  
. 鐗涗汉浜戦泦,涓€浜╀笁鍒嗗湴

**According to this, write a naive recursion implementation. Walk them through. Make sure to illustrate the idea clearly!!!**  
  
def is\_match\_recursion(s, p):  
    if not s: # Corner case.  
        return p.count('\*') == len(p). from: [1point3acres.com/bbs](http://1point3acres.com/bbs)   
  
    if not p: # Corner case.  
        return not s  
  
    if p[-1] == '?' or s[-1] == p[-1]: # Recursive call.  
        return is\_match(s[:-1], p[:-1])  
  
    if p[-1] == '\*’: # Recursive call.  
        return is\_match(s, p[:-1]) or is\_match(s[:-1], p)  
  
    return False  
  
. more info on [1point3acres.com](http://1point3acres.com/)  
Two [opt](http://tinyurl.com/mwuwuhv" \t "_blank)imizations:

1. Change passing str to passing list + indices, save memory in recursive calls.
2. Avoid duplicated computation by importing memorization {(si, pi): True/False}. 1point 3acres 璁哄潧

. more info on [1point3acres.com](http://1point3acres.com/)  
. more info on [1point3acres.com](http://1point3acres.com/)

-google 1point3acres  
  
**Extend the idea above to Dynamic Programming. Make sure to be clear about the tabular fill-up (relationship), and corner cases.**. 鍥磋鎴戜滑@1point 3 acres  
  
def is\_match(s, p):  
    m = len(s)  
    n = len(p)  
  
    matched = [[False] \* (n+1) for \_ in xrange(m+1)]  
    matched[-1][-1] = True # Corner Case.  
    for i in xrange(n): # Corner Case.  
        if p*!= '\*':  
            break  
        matched[-1] = True  
  
    for si in xrange(m):. 涓€浜�-涓夊垎-鍦帮紝鐙鍙戝竷  
        for pi in xrange(n):. 鍥磋鎴戜滑@1point 3 acres  
            if s[si] == p[pi] or p[pi] == '?’:  # Relationship.  
                matched[si][pi] = matched[si-1][pi-1]  
            elif p[pi] == '\*’:                         # Relationship.  
                matched[si][pi] = matched[si][pi-1] or matched[si-1][pi]  
            else:  
                matched[si][pi] = False  
    return matched[m-1][n-1]*

*鏉ユ簮涓€浜�.涓夊垎鍦拌鍧�.   
.1point3acres缃�****Followup, Backtracking!****.*[*1point3acres.com/bbs*](http://1point3acres.com/bbs) *si, pi = 0, 0  
last\_si, last\_pi = None, None  
  
while si < len(s):.鏈枃鍘熷垱鑷�1point3acres璁哄潧  
    if pi < len(p) and (s[si] == p[pi] or p[pi] == '?'):  
        si += 1  
        pi += 1. from:*[*1point3acres.com/bbs*](http://1point3acres.com/bbs)*elif pi < len(p) and p[pi] == '\*':  
        last\_si = si + 1. 1point 3acres 璁哄潧  
        last\_pi = pi + 1  
        pi += 1  
    elif last\_pi:. from:*[*1point3acres.com/bbs*](http://1point3acres.com/bbs)*pi = last\_pi  
        si = last\_si  
        last\_si += 1  
    else:  
        return False  
  
  
  
return p[pi:].count('\*') == len(p) - pi. 1point 3acres 璁哄潧****Second set*** *First round:  
Power of 4****Of course it could be solved by brute force= =***

*while abs(num) > 1:  
    if num % 4 != 0: return False. Waral 鍗氬鏈夋洿澶氭枃绔�,  
    num /= 4  
return num == 1*

***It reminds me of power of 2****:*

*num & (num-1) == 0 # power of 2*

*This means there’s only one bit in binary representation.  
  
All we need to do is to add another condition.*

*- num > 0 and num | 0x55555555 == 0x55555555 # explain 0x55555555  
  
- (num & 0x55555555) != 0  
- (num-1) % 3 == 0 # 3n + 1*

*two's complement, explain negative numbers  
.1point3acres缃�  
  
. Waral 鍗氬鏈夋洿澶氭枃绔�,  
Random iterator dividable by 5****Be clear what to remove and when to remove them!***

*class Iterator(object):  
    def \_\_init\_\_(self, nums):. more info on*[*1point3acres.com*](http://1point3acres.com/) *self.nums = nums  
        self.x = 0.鐣欏璁哄潧-涓€浜�-涓夊垎鍦�  
        self.y = 0.1point3acres缃�  
        self.last\_return = None  
  
    def hasNext(self):.鐣欏璁哄潧-涓€浜�-涓夊垎鍦�  
        while self.x < len(self.nums) and \  
              self.y >= len(self.nums[self.x]):  
            self.x += 1  
            self.y = 0  
        return self.x < len(self.nums) and self.y < len(self.nums[self.x])  
  
    def next(self):  
        if self.hasNext():  
            val = self.nums[self.x][self.y]  
            self.last\_return = (self.x, self.y)             
            self.y += 1  
            return val  
        else:.1point3acres缃�  
            raise Exception('No more values').1point3acres缃�  
  
    def remove(self):  
        if self.last\_return:  
            x, y = self.last\_return. From 1point 3acres bbs  
            self.last\_return = None  
            self.nums[x].pop(y)  
            self.x, self.y = x, y  
        else:  
            raise Exception('None couldn\'t be removed')*

***Mod 5 iterator.***

*bool hasNext() {  
        # Either way is okay, hasNext() or next().鏈枃鍘熷垱鑷�1point3acres璁哄潧****while (self.n % 5 != 0 and Iterator::hasNext()) {******self.n = Iterator::hasNext();******}*** *return self.n % == 0;  
    }  
  
int next() {. 鐗涗汉浜戦泦,涓€浜╀笁鍒嗗湴  
        if (!hasNext()). visit*[*1point3acres.com*](http://1point3acres.com/)*for more.  
            throw runtime\_error("no more elements!");  
  
        val = self.n  
. from:*[*1point3acres.com/bbs*](http://1point3acres.com/bbs)*self.n = INIT\_VAL # not equal to n \* 5. 1point 3acres 璁哄潧  
  
        return val;  
    }*

*Some theory. 鍥磋鎴戜滑@1point 3 acres  
- 理论。给两个set，如果取他们的交集，问了各个方法的优缺点（时间复杂度+空间复杂度）*

*- brute force  
- sort one, binary search the other one  
- sort, then two pointers  
- hashmap  
~~- merge，归并排序duplicates???~~  
.鐣欏璁哄潧-涓€浜�-涓夊垎鍦�*

*- 理论。各种sorting algo 的优缺点。  
.鏈枃鍘熷垱鑷�1point3acres璁哄潧*

*- bubble, insertion  
- bucket sort  
- shell   
- merge sort  
- quick sort*

*Second round. Waral 鍗氬鏈夋洿澶氭枃绔�,  
Game of life****This is 1D solution.***

*def game\_of\_life\_1d(arr):  
    n = len(arr)  
. more info on*[*1point3acres.com*](http://1point3acres.com/) *for x in xrange(n):  
        live\_count = 0  
        if arr[(x+1+n) % n] & 1 == 1:  
            live\_count += 1  
        if arr[(x-1+n) % n] & 1 == 1:  
            live\_count += 1  
  
        if live\_count == 1:  
            arr[x] \*= (arr[x] + 1) % 2 \* 2. 1point 3acres 璁哄潧  
        else:. 1point 3acres 璁哄潧  
            arr[x] += arr[x] \* 2  
  
    for x in xrange(n):  
        arr[x] >>= 1*

*. visit*[*1point3acres.com*](http://1point3acres.com/)*for more.****This is 2D solution. Ask about cross border situation.***

*def game\_of\_life\_2d(grid):  
    m, n = len(grid), len(grid[0]). 鐗涗汉浜戦泦,涓€浜╀笁鍒嗗湴  
  
    for x in xrange(m):  
        for y in xrange(n):. 鐗涗汉浜戦泦,涓€浜╀笁鍒嗗湴  
            live\_count = get\_live\_count(grid, x, y, m, n). more info on*[*1point3acres.com*](http://1point3acres.com/) *#  print x, y, live\_count  
            if grid[x][y] == 1:  
                if live\_count == 2 or live\_count == 3:.1point3acres缃�  
                    grid[x][y] += 2  
            elif live\_count == 3: 鏉ユ簮涓€浜�.涓夊垎鍦拌鍧�.   
                grid[x][y] += 2  
  
    for x in xrange(m):. 涓€浜�-涓夊垎-鍦帮紝鐙鍙戝竷  
        for y in xrange(n):  
            grid[x][y] >>= 1  
  
  
# Ask. Be clear about this.  
def get\_live\_count(grid, x, y, m, n):  
    live\_count = 0  
    if grid[x][(y+1+n) % n] & 1 == 1:  
        live\_count += 1. from:*[*1point3acres.com/bbs*](http://1point3acres.com/bbs)*if grid[x][(y-1+n) % n] & 1 == 1:  
        live\_count += 1  
  
    if grid[(x+1+m) % m][y] & 1 == 1:  
        live\_count += 1  
    if grid[(x-1+m) % m][y] & 1 == 1:. From 1point 3acres bbs  
        live\_count += 1. from:*[*1point3acres.com/bbs*](http://1point3acres.com/bbs)*. visit*[*1point3acres.com*](http://1point3acres.com/)*for more.  
    if grid[(x+1+m) % m][(y+1+n) % n] & 1 == 1:  
        live\_count += 1  
    if grid[(x+1+m) % m][(y-1+n) % n] & 1 == 1:  
        live\_count += 1  
  
    if grid[(x-1+m) % m][(y+1+n) % n] & 1 == 1:  
        live\_count += 1  
    if grid[(x-1+m) % m][(y-1+n) % n] & 1 == 1:  
        live\_count += 1.鏈枃鍘熷垱鑷�1point3acres璁哄潧  
  
    return live\_count*

*Followup:  
- infinite grid (no border!)*

*cur\_generation = {(i, j) for i, row in enumerate(board) for j, live in enumerate(row) if live}*

*ctr = collections.Counter((I, J)  
                                         for i, j in live  
                                         for I in range(i-1, i+2)  
                                         for J in range(j-1, j+2)  
                                         if I != i or J != j)*

*next\_generation = {ij for ij in ctr if ctr[ij] == 3 or ctr[ij] == 2 and ij in live}*

*- really large grid!*

*- multi-thread / multi-process: no problem!*

*- map/reduce: split into squares, pass them with their own border, so that they don’t have to use internet to get data.*

*. Waral 鍗氬鏈夋洿澶氭枃绔�,  
  
Text Editor oo design  
Rope(data structure). 鐗涗汉浜戦泦,涓€浜╀笁鍒嗗湴  
  
我一开始说要用char[][]。 但是給批评了好久...最后用了arraylist<arrayList<String>> 貌似...勉强通过了....然后...用什么储存..hightlight的...我一开始用hashset了..但是有个问题就是.delete的东西里面.刚好是hightlight的..那你怎么知道..你delete 的东西..也在hashset里面...所以.后来我用interval tree了.....他说...有道理多了....然后最后问了redo 和undo 怎么做...我用两个stack 储存operation.....  
. 涓€浜�-涓夊垎-鍦帮紝鐙鍙戝竷  
需要实现分页，编辑，以及undo和redo。  
  
  
用什么数据结构来存这些输入进来的字符最有效。最后给出的是 list + array 的方法。。。  
  
. 鐗涗汉浜戦泦,涓€浜╀笁鍒嗗湴  
text editor，我先给了一个array的solution。后来时间不太多了，问我有没有更好的，我就说用binary tree，其实就是rope的那个实现。结构框架都和各个方法怎么来实现也都说了个大概。  
  
  
insert(p), delete(p1, p2), highlight(p1, p2)，redo/undo, save/load update, search  
text editor需要insert，remove，highlight，需要想办法去index每次插入的object，原po说的interval tree应该就是index的方式吧。  
关键点在于text打算怎么存  
store highlight?  
他要求三天后再load这个text,需要可以undo三天前的操作. save的时候 保存成xml类型之类的 把之前的操作也一起存下来. 鐗涗汉浜戦泦,涓€浜╀笁鍒嗗湴  
  
  
- What objects?   
- Larget string.  
  
- What ds?  
- Rope  
  
- What operation?  
- Insert, delete, search, print <- index, concat, split  
  
- What about highlight?  
.鏈枃鍘熷垱鑷�1point3acres璁哄潧  
- add field ‘font’ in the internal nodes, split when necessary  
. From 1point 3acres bbs  
- Undo/redo?  
  
- Use two stacks, one for each, and make operations into objects  
  
  
Third round.鐣欏璁哄潧-涓€浜�-涓夊垎鍦�  
Debug guava  
.鐣欏璁哄潧-涓€浜�-涓夊垎鍦�  
  
  
At least 4 functions need to rewrite, debug:  
- put  
- size  
- clear  
- clearall*

*// put  
  public boolean put(@Nullable K key, @Nullable V value) {  
    Collection<V> collection = map.get(key);  
    if (collection == null) {  
      collection = createCollection(key);  
      if (collection.add(value)) {  
        totalSize++;  
        map.put(key, collection);  
        return true;  
      } else {  
        throw new AssertionError("New Collection violated the Collection spec");  
      }.鏈枃鍘熷垱鑷�1point3acres璁哄潧  
    } else if (collection.add(value)) {.1point3acres缃�  
      totalSize++;.鐣欏璁哄潧-涓€浜�-涓夊垎鍦�  
      return true;  
    } else {  
      return false;  
    }  
  }  
  
  /\*\*  
   \* Removes all values for the provided key.  
   \*/  
  private void removeValuesForKey(Object key) {.鏈枃鍘熷垱鑷�1point3acres璁哄潧  
    Collection<V> collection = Maps.safeRemove(map, key);  
  
    if (collection != null) {. visit*[*1point3acres.com*](http://1point3acres.com/)*for more.  
      int count = collection.size();  
      collection.clear();  
      totalSize -= count;  
    }  
  }  
. Waral 鍗氬鏈夋洿澶氭枃绔�,  
  // get  
  public Collection<V> get(@Nullable K key) {  
    Collection<V> collection = map.get(key);  
    if (collection == null) {  
      collection = createCollection(key);  
    }.*[*1point3acres.com/bbs*](http://1point3acres.com/bbs) *return wrapCollection(key, collection);  
  }. 鍥磋鎴戜滑@1point 3 acres*

*. 涓€浜�-涓夊垎-鍦帮紝鐙鍙戝竷****Third Set*** *First round  
LRU  
Get  
- illegal input  
- update (key, value)  
  
Set  
- key existed — update  
- new key  
— enough space  
— not enough space  
  
Size  
  
Fast  
- in get, hash table to get value and move it to head  
- in set, double linked list for quick modification  
— move to head 鏉ユ簮涓€浜�.涓夊垎鍦拌鍧�.   
— add to head. 1point 3acres 璁哄潧  
— delete from tail  
. 鐣欏鐢宠璁哄潧-涓€浜╀笁鍒嗗湴  
  
Second round  
Best Time to Buy and Sell Stock III****One dimension*** *- max\_profit = array(n)  
- max\_profit = max(max\_profit, # day i is not involved in transaction, we’re using stat from previous day, or  
                                  prices - min\_prices from [0 .. i-1] # stock sold on day i, buy on j from [0 .. i-1], max\_minus\_prices from [0 .. i-1])****K dimension (k transactions)****.鏈枃鍘熷垱鑷�1point3acres璁哄潧  
- max\_profit = matrix[k \* n], max profit till day i using UP to k transactions.  
- max\_profit[kk] = max(max\_profit[kk][i-1], # same, not involved with day i  
                                        max(prices - prices[j] + max\_profit[kk-1][j]) # One transaction bought on day j, sold on day i.鐣欏璁哄潧-涓€浜�-涓夊垎鍦�  
.鐣欏璁哄潧-涓€浜�-涓夊垎鍦�  
                                                                                                               # plus max\_profit[kk-1][j], on day j using k - 1 transactions   
  
- where the underline part is computed multiple times, each time we can just use max\_diff = max(max\_diff, - prices[j] + max\_profit[kk-1][j]) to save time.  
  
  
Median of two sorted arrays  
  
Test cases:  
    def test\_empty\_array(self):  
        self.assertEqual(self.solution.findMedianSortedArrays([], [1]), 1)  
        self.assertAlmostEqual(  
            self.solution.findMedianSortedArrays([], [1, 2]), 1.5)-google 1point3acres  
        self.assertEqual(self.solution.findMedianSortedArrays([2], []), 2)  
        self.assertEqual(self.solution.findMedianSortedArrays([1, 3], []), 2). visit*[*1point3acres.com*](http://1point3acres.com/)*for more.  
  
    def test\_three\_elements\_left(self):  
        self.assertEqual(self.solution.findMedianSortedArrays([2, 3], [1]), 2)  
        self.assertEqual(self.solution.findMedianSortedArrays([1, 3], [2]), 2)  
  
    def test\_four\_elements\_left(self):  
        self.assertEqual(  
            self.solution.findMedianSortedArrays([2, 5], [1, 4]), 3)  
        self.assertEqual(. 鍥磋鎴戜滑@1point 3 acres  
            self.solution.findMedianSortedArrays([4, 5], [1, 2]), 3)  
  
    def test\_different\_lengths(self):  
        self.assertEqual(self.solution.findMedianSortedArrays(  
            [1, 2, 2, 4, 5], [1, 4]), 2)  
        self.assertEqual(self.solution.findMedianSortedArrays(  
            [1, 2, 4, 5], [1, 2]), 2)  
        self.assertEqual(self.solution.findMedianSortedArrays(  
            [1, 2, 2, 4, 5], [1, 4, 5]), 3)  
        self.assertEqual(self.solution.findMedianSortedArrays(  
            [1, 2, 4, 5], [1, 2, 7]), 2)  
  
    def test\_two\_median\_relationship(self):. From 1point 3acres bbs  
        self.assertEqual(self.solution.findMedianSortedArrays(  
            [1, 2, 2, 4, 5], [1, 4, 5]), 3)  
        self.assertEqual(self.solution.findMedianSortedArrays(  
            [1, 2, 2, 4, 5], [1, 1, 5]), 2). from:*[*1point3acres.com/bbs*](http://1point3acres.com/bbs)*self.assertEqual(self.solution.findMedianSortedArrays(  
  
            [1, 2, 2, 4, 5], [1, 2, 5]), 2)  
  
Algorithm:  
    def find\_kth\_smallest(self, nums1, nums2, k):. more info on*[*1point3acres.com*](http://1point3acres.com/) *m, n = len(nums1), len(nums2). 鐣欏鐢宠璁哄潧-涓€浜╀笁鍒嗗湴  
        if m > n:.鐣欏璁哄潧-涓€浜�-涓夊垎鍦�  
            return self.find\_kth\_smallest(nums2, nums1, k) 鏉ユ簮涓€浜�.涓夊垎鍦拌鍧�.   
  
        if not nums1:  
            return nums2[k - 1]  
  
        if k == 1:.1point3acres缃�  
            return min(nums1[0], nums2[0])  
  
        i = min(k / 2, m)  
        j = min(k / 2, n)  
        if nums1[i - 1] > nums2[j - 1]:  
            return self.find\_kth\_smallest(nums1, nums2[j:], k - j)  
        else:  
            return self.find\_kth\_smallest(nums1[i:], nums2, k - i)  
  
    def findMedianSortedArrays\_v2(self, A, B):  
        m, n = len(A), len(B)  
        if m > n:  
            A, B, m, n = B, A, n, m  
        if n == 0:  
            raise ValueError  
  
        imin, imax, half\_len = 0, m, (m + n + 1) / 2 鏉ユ簮涓€浜�.涓夊垎鍦拌鍧�.   
        while imin <= imax:  
            i = (imin + imax) / 2  
            j = half\_len - i  
            if i < m and B[j - 1] > A:  
                # i is too small, must increase it  
                imin = i + 1  
            elif i > 0 and A[i - 1] > B[j]:  
                # i is too big, must decrease it. 1point 3acres 璁哄潧  
                imax = i - 1  
            else:  
                # i is perfect. visit*[*1point3acres.com*](http://1point3acres.com/)*for more.  
  
                if i == 0:.鏈枃鍘熷垱鑷�1point3acres璁哄潧  
                    max\_of\_left = B[j - 1].鐣欏璁哄潧-涓€浜�-涓夊垎鍦�  
                elif j == 0:  
                    max\_of\_left = A[i - 1].*[*1point3acres.com/bbs*](http://1point3acres.com/bbs) *else:  
                    max\_of\_left = max(A[i - 1], B[j - 1]). 鐣欏鐢宠璁哄潧-涓€浜╀笁鍒嗗湴  
  
                if (m + n) % 2 == 1:  
                    return max\_of\_left  
  
                if i == m:. 1point 3acres 璁哄潧  
                    min\_of\_right = B[j]  
                elif j == n:  
                    min\_of\_right = A  
                else:  
                    min\_of\_right = min(A, B[j])  
  
                return (max\_of\_left + min\_of\_right) / 2.0  
  
  
Third round  
ATM oo design. 1point 3acres 璁哄潧  
- 设计interface  
- 有基本的interface, 写implementation*

*.鐣欏璁哄潧-涓€浜�-涓夊垎鍦�  
Accounts:  
- checking. 鍥磋鎴戜滑@1point 3 acres  
- saving  
.鏈枃鍘熷垱鑷�1point3acres璁哄潧  
Functions:  
- withdraw. from:*[*1point3acres.com/bbs*](http://1point3acres.com/bbs)*- deposit  
- transfer  
- inquiry  
- setting  
  
class ATM.1point3acres缃�  
class User  
class Accounts*