

First Onsite :

Fundamental CS, Design, Problem_solving

Single-thread, multi-thread knowledge, concurrency, race condition, deadlock, debug

1, 定义一个Buddy System为complete binary tree, 一个node可能为0或者1, 当且仅当所有child的值为1的时候node值为1; (a complete binary tree is a binary tree in every level, except possibly the last, is completely filled, and all nodes are as far left as possible)
Given a complete binary tree with nodes of values of either 1 or 0, the following rules always hold:

(1) a node's value is 1 if and only if all its subtree nodes' values are 1

(2) a leaf node can have value either 1 or 0

Implement the following 2 APIs:

set_bit(offset, length), set the bits at range from offset to offset+length-1

clear_bit(offset, length), clear the bits at range from offset to offset+length-1

i.e. The tree is like:

```
      0
     /\
    0 1
   /\ /\
  1 0 1 1
 /\ /\ /\
1 1 1 0 1
```

Since it's complete binary tree, the nodes can be stored in an array:

[0,0,1,1,0,1,1,1,1,0,1]

从offset开始到offset+len-1置为1或0, 但是同时也会影响到上面的元素, 所以要迭代处理。

BuddyBitmapSystem(alienware) (offset并不只限于最后一行)

假设bit存储在bits[level][number]里(offset从最后一层开始): 考虑一层一层的改, 而不是由下改到顶点;

3, Draw a circle.

https://www.tutorialspoint.com/computer_graphics/circle_generation_algorithm.htm

<https://www.cs.uic.edu/~jbell/CourseNotes/ComputerGraphics/Circles.html>

<http://groups.csail.mit.edu/graphics/classes/6.837/F98/Lecture6/circle.html>

brute force:

```
public void drawCircle(int x, int y, int r){
    for(int i = -r; i <= r; i++){
        for(int j = -r; j <= r; j++){
            if(i * i + j * j <= r * r){
                setPixel(x + i, y + j);
            }
        }
    }
}
```

python(1/8 and reflect):

```
from sets import Set
def draw_circle(r2):
    result = Set([])
```

```

x = 1
y = 0
while x*x <= r2:
    for y in range(x+1):
        if x*x+y*y == r2:
            result.update(Set([(x,y),(x,-y),(-x,-y),(-x,y),(y,x),(y,-x),(-y,-x),(-y,x)]))
    x+=1
return result

def draw_circle_bi_search(r2):
    result = Set([])
    x = 1
    y = 0
    while x*x <= r2:
        y_start = 0
        y_end = x
        while y_start <= y_end:
            y_mid = y_start+(y_end-y_start)/2
            if x*x + y_mid*y_mid == r2:
                result.update(Set([(x,y_mid),(x,-y_mid),(-x,-y_mid),(-x,y_mid),(y_mid,x),(y_mid,-x),(-y_mid,-x),(-y_mid,x)]))
                break
            elif x*x + y_mid*y_mid < r2:
                y_start = y_mid+1
            else:
                y_end = y_mid-1

        x+=1
    return result

```

follow up :

1. 如何高效地计算像素点的位置？
2. 是否可以只计圆上一部分像素的位置，其余像素用极其简单的数学计算得到？（提示：四个象限，1/8）
3. x坐标是否每次都加1？y坐标是否每次都加1？为什么？
4. 当一圆上的一个点的理论值不是整数时，要如何取整？舍去？补足？还是四舍五入？
5. 如何不用开根号sart()函数？
6. 如何不用取绝对值？（提示：理论值是否一定在两个整数之间？）
7. 下一次画点的像素点候选位置有几个？
8. 下一次画点的像素点的候选位置可否根据上一次的点计算出来？
9. 下一次画点的像素点的候选位置可否不用求平方就直接根据上一次的点计算出来？
10. 如果r很小，比如r == 3，该如何多画几个点？

3，设计一个task dispatching system，里面有一个task queue和两个function。

trigger 。这个function运行并清空task queue中所有的tasks 。

addTask 。在trigger之前把task加入task queue ，在trigger之后直接运行task 。

```
trigger(){
    acquire(mutex_a);
    a = true;
    dispatch all tasks in the queue;
    head = null;
    tail = null;
    release(mutex_a);
}
```

```
addtask(){
    acquire(mutex_a);
    if(a)
        dispatch the task;
    else{
        if(head == null){
            head = task;
            tail = task;
        } else{
            tail.next = task;
            tail = task;
        }
    }
    release(mutex_a);
}
```

和Callback ， fire相似 ： <https://instant.1point3acres.com/thread/177053>

4 ， LRU cache ， Skyline ， Permutation, combination sum ；

5 ， O (1) Map ： Global version <http://massivealgorithms.blogspot.com/2016/05/constant-clear-map.html> / / <https://aonecode.com/getArticle/12>

设计一个Map<Integer, Integer> ，满足下面的复杂度：

add: O(1) deletion: O(1) lookup: O(1) clear:O(1) iterate: O(number of elements) 。

Solution:

By using a dictionary, we can meet all the above requirement except for the clear operation. The clear operation requires us to clear all content in the map in O(1) time. The only way to do this is to use a timestamp and mark each key/value pair. When calling clear(), increment the timestamp. So if we look up the map after clearing it, the older key/value pair would not be considered (假清空，有没有其他办法？)

exist.

Details:

1, Map contains a dictionary<int, node> and int currVersion.

2, Node contains int value and int version.

3, Add: if the key is not in dict, create new node with currVersion and value and insert <key, node>.

If the key is in the dict but its version is older than currVersion, remove the existing pair then add new pair. If the key is in the dict and its version is same as currVersion, throw exception.

4, Delete: if the key is not in the dict, return false. If the key is in the dict but its version is older than currVersion, return false. Otherwise, remove the pair.

5, GetValue: Same as Delete.

6, Clear: currVersion++. Thus all existing key/node pair become invalid. Time: O(1)

```
public class Node{
    public int value{get;set;}
    public int version{get;set;} //time stamp
    public Node(int va, int ve){
        value = va;
        version = ve;
    }
}
```

```
public class MyMap{
    private Dictionary<int,Node> dict;
    private int currVersion;
    public MyMap(){
        dict = new Dictionary<int,Node>();
        currVersion = 0;
    }
}
```

//add key value pair into map, note if there is a same key with older version

//first delete older key, then add the new key value pair

```
public void Add(int key, int value){
    if(dict.ContainsKey(key)){
        if(dict[key].version < currVersion)
            dict.Remove(key);
        else
            throw new ArgumentException("duplicate key!");
    }
    //if there is no same key or there is a same key with older version,
    //we always need to add the key to dict
    Node temp = new Node(value,currVersion);
    dict.Add(key,temp);
}
```

//if delete successfully, return true, otherwise, return false

```
public bool Delete(int key){
    if(!dict.ContainsKey(key))
        return false;
    if(dict[key].version < currVersion)
        return false; // also remove the old one
    dict.Remove(key);
    return true;
}
```

```
//get value using key
public int GetValue(int key){
    if(!dict.ContainsKey(key))
        throw new ArgumentException("key doesn't exist!");
    if(dict[key].version < currVersion)
        throw new ArgumentException("key doesn't exist!");
    return dict[key].value;
}

//clear the whole map
public void Clear(){
    currVersion++;
}
}
```

HashMap 和 arraylist 解法 

Solution is 2 arrays, map, list where for map, the index is the data x, the value is its index in list:
 map = 0 if you first add a 2 to list. list stores the real value of the data remove is from map you
 find the index in list, you swap the list with list[map] for clear, you just set size to be 0 to lookup
 an item x, map

6.What data structure would you use to construct a skip list? Implement search() and insert().

<http://www.cnblogs.com/binyue/p/4545555.html>

<http://www.mathcs.emory.edu/~cheung/Courses/323/Syllabus/Map/skip-list-impl.html>

<http://www.sanfoundry.com/java-program-implement-skip-list/>

<http://m.2cto.com/kf/201702/599228.html>

7，判断四个点能否组成正方形。（及正方形的个数）

<http://www.it610.com/article/5366872.htm>

8，align rectangle

<https://instant.1point3acres.com/thread/190646>

9，image smoother（先加列再加行）

```
public void rowAdd(int[][] matrix){
    if(matrix == null) return;
    int temp0 = 0;
    int temp1 = 0;
    for(int i = 0; i < matrix.length; i++){
        for(int j = 0; j < matrix[0].length; j++){
            if(j == 0){
                temp0 = matrix[i][j];
                matrix[i][j] += matrix[i][j + 1];
            }else if(j == matrix[0].length - 1){
                if(j % 2 == 0){
                    matrix[i][j] += temp1;
                }
            }
        }
    }
}
```

```

        }else if(j % 2 == 1){
            matrix[i][j] += temp0;
        }
    }
    if(j > 0 && j < matrix[0].length - 1 && j % 2 == 0){
        temp0 = matrix[i][j];
        matrix[i][j] += temp1;
        matrix[i][j] += matrix[i][j + 1];
    }else if(j > 0 && j < matrix[0].length - 1 && j % 2 == 1){
        temp1 = matrix[i][j];
        matrix[i][j] += temp0;
        matrix[i][j] += matrix[i][j + 1];
    }
}
}
}

```

10，一个disk有很多chunks，但只有三种类型称为A，B，C。要求把乱序的disk chunks重排为[A A B B B B C C]，所有A在前面，C在后面，B在中间，只能通过1) read某个index的chunk 2) swap两个chunk的data 这两个操作来实现。follow up是尽量减少2) 的次数

11，Reverse file。

2017(1-3月) [码农类 硕士 全职@PureStorage](#) - 内推 - [Onsite](#) | [Fail](#)在职跳槽

2月份朋友内推，被我出去旅游忘掉了，但是他们家HR都非常友好，3月份还找了我，我就做了OA，12题版，地里有全套，我就不说了，但是地里的答案是有问题的，我记得有两个错，一定要认真自己看，他们家OA不全对是不会onsite的；好的，昨天我第一次onsite了；

第一次onsite应该都是四大题不会变：O (1) Map， Draw a circle， BuddyBitMap， Callback (register () and fire ()) ；

我被问到的是BuddyBitMap和Callback；

注意BuddyBitMap里的complete binary tree是用bits[level][index]这样的方式给出的，然后有start， len来表示被更改的子树的位置和范围，height表示树的高度；从最后一层开始；

一定要注意时间复杂度和空间复杂度，如果不是最优他会让你一直改一直改；还有要想出一层一层的改的方法，不要从低端backtrace回去；而且要快要熟练，我犹豫了两下，换了两种方法估计就pass了；

callback没什么好说的，也是要熟练，他们家其实不算难，难在面试官的 follow up 在于考你是不是真的理解多线程和thread—safe这些，假设多种情况问你这样是安全的吗那样是安全的吗，会发生什么样的情况等等；

今天已经打电话拒了也是非常高效率；

我把我自己总结的面经放在下面攒点人品吧，希望大家面他家的时候能更顺利！

另外各位有什么内推的机会给我好不好！给一个女生做码农的机会，给世界一点机会！谢谢了！