## Amazon

|  |
| --- |
| 给一个BST，一个range,，求树上在这个range里最小的那个node |

遍历树，range为【min,max】，1)node比min小，go right. 2)node在范围内go left，3）node比max大，go left

public int calsmallest(TreeNode node,int min,int max){  
    int minValue = Integer.MAX\_VALUE;  
    while(node!=null){. 鐗涗汉浜戦泦,涓€浜╀笁鍒嗗湴  
        if(node.val>=min&&node.val<=max){  
            minValue = Math.min(node.val,minValue);  
            node=node.left;  
        }else if(node.val<min) node=node.right;  
        else node=node.left;  
    }  
    return minValue;  
}

2：

|  |
| --- |
| 写一个 method MovingAverage(int num) 提前告诉你一个int n 每次调用这个方法 就把num加进一个队列中 然后返回最近n个数的平均值 鏉ユ簮涓€浜�.涓夊垎鍦拌鍧�.  比如 如果n是3那么 MovingAverage(1)->1/1 MovingAverage(2)->(1+2)/2 MovingAverage(3)->(1+2+3)/3 MovingAverage(4)->(2+3+4)/3 MovingAverage(5)->(3+4+5)/3 |

维护一个大小为n的额外空间，每次往队列里加数时，也往这个额外空间加数，超过了就queue pop再加

第二轮第1题用滑动窗口的方法就行了吧？进窗口的时候，sum += A[end++], 出窗口的时候 sum -= A[begin++], 空间复杂度)(1)

|  |
| --- |
| 第二题 Number of islands ii leetcode原题 |

Facebook

|  |
| --- |
| <https://leetcode.com/problems/intersection-of-two-arrays/description/>  <https://leetcode.com/problems/intersection-of-two-arrays-ii/description/> |

|  |
| --- |
| 2. 似乎是新题， 给一个排序好的数组，数一数有多少个子集（subset）使得子集里面的最大元素加上最小元素小于K。 A: [2, 3, 5, 7]  K: 8 . 鐣欏鐢宠璁哄潧-涓€浜╀笁鍒嗗湴 => # of subsets S: Max(S) + Min(S) < K => [2] , [2, 3], [2, 5], [2, 3, 5], [3] => #: 5 |

排好序，然后two pointer, 当找到 a*+ a[j]  < k,  && i < j 的时候，包含i的子集有 2^(j-i) 个，他们都满足条件。 然后i++，继续看，否则j--， O（n）就够了*

*while (i < j)  
   if (a + a[j] < k)   count += Math.pow(2, j-i); i++. 鍥磋鎴戜滑@1point 3 acres  
   else j--*

1. public int subarraySum(int[] nums, int K) {
2. if (nums == null || nums.length == 0 || nums[0] \* 2 >= K) {
3. return 0;
4. }. Waral 鍗氬鏈夋洿澶氭枃绔�,
5. int sum = 0, i = 0, j = nums.length - 1;
6. while (i <= j) {
7. if (nums[i] + nums[j] < K) {
8. sum += (int)Math.pow(2, j - i); 鏉ユ簮涓€浜�.涓夊垎鍦拌鍧�.
9. i++;
10. } else {
11. j--;
12. }
13. }
14. return sum;. 鐗涗汉浜戦泦,涓€浜╀笁鍒嗗湴
15. }

这题感觉是subset变形。第二题求所有最大windows, 然后计算子集而已

1 2 3 8的 1+8<10的时候 2^3 = 8是有8个subset 1, 1 2, 1 3, 1 8, 1 2 3, 1 2 8, 1 3 8, 1 2 3 8  
然后下一个2 + 3<10 的时候 2个解 2， 2 3  
然后是 3 + 3 <10 的时候 3 一个解  
一共11个subsets

呃，two pointers依然可以做，实际上就是找出所有的满足(x, y), 其中x + y < K的数对就行了，因为数组是排序过的，所以找出这些pair之后，就可以算出，中间有多少数，然后subset的个数就是2^(ix - iy - 1)个

要考虑一个元素时候的特殊情况

backtrack + dfs解法，算是subset的变种题把  
void dfs(vector<int> nums, int idx, int &res, vector<int> &cur, int target)  
{  
    if(idx == nums.size())  
    {  
        return;  
    }  
    else  
    {.1point3acres缃�  
        for(int i = idx; i < nums.size(); i++)  
        {  
            cur.push\_back(nums*);  
            if((cur.size() == 1 && cur[0] < target) || (cur[0] + cur.back() < target))  
            {  
                for(auto it : cur)  
                {  
                    cout<<it<<" ";  
                }. more info on*[*1point3acres.com*](http://1point3acres.com/) *cout<<endl;.1point3acres缃�  
                res++;  
                dfs(nums, i + 1, res, cur, target);  
            }  
            cur.pop\_back();  
            while(i < nums.size() - 1 && nums == nums[i + 1])i++;  
        } 鏉ユ簮涓€浜�.涓夊垎鍦拌鍧�.   
    }.鐣欏璁哄潧-涓€浜�-涓夊垎鍦�  
}  
int count(vector<int> nums, int target)  
{  
    int res = 0;. 鐣欏鐢宠璁哄潧-涓€浜╀笁鍒嗗湴  
    vector<int> cur;. Waral 鍗氬鏈夋洿澶氭枃绔�,  
    dfs(nums, 0, res, cur, target);  
    return res;  
}  
  
int main()  
{  
    vector<int> nums = {2, 2, 3, 5, 7};. [1point3acres.com/bbs](http://1point3acres.com/bbs)  
    cout<<count(nums, 8)<<endl;  
      
}  
. 鐗涗汉浜戦泦,涓€浜╀笁鍒嗗湴****补充内容 (2017-8-4 10:30):*** *if condition里(cur.size() == 1 && cur[0] < target) 条件去掉才对*