刷题笔记

# Binary Search

* 主要参考的模板来自youtube视频： https://www.youtube.com/watch?v=25086D5uZmY&t=829s

## 1. 标准二分查找:

1. 判断条件是 l<=h

2. 每次都是更新l和h，并且是 +1 和 -1

3. 每次在判断==之后，的if不要加else，因为如果是复杂条件判断的话，每次都要更新l和h

1. **int** binarySearch(**int**[] nums, **int** target){
2. l = 0, r = nums.length-1;
3. **while**(l <= h){
4. **int** mid = l  + (h - l) /2;
5. **if**(nums[mid] == target){
6. **return** mid;
7. }
9. **if**(nums[mid] < target){
10. l = mid + 1;
11. }**else** {
12. h = mid -1;
13. }
14. }
16. **return** -1;
17. }

* 典型题目：

[33.Search in Rotated Sorted Array](https://leetcode.com/problems/search-in-rotated-sorted-array)

[81.Search in Rotated Sorted Array II](https://leetcode.com/problems/search-in-rotated-sorted-array-ii)

[34. Find First and Last Position of Element in Sorted Array](https://leetcode.com/problems/find-first-and-last-position-of-element-in-sorted-array/)

## 2. 变形查找, 参考视频里的解说,利用谓词P来扩展判断条件,查找第一个yes的代码

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

1. 判断条件是l < h
2. 每次是l+1 更新,而h直接等于mid
3. While之后要有complain的判断
4. **public** **int** searchInsert(**int**[] nums, **int** target) {
5. **if**(nums.length == 0) **return** 0;
7. // find the first >= target
8. **int** l = 0, h = nums.length-1;
9. **while**(l < h){
10. **int** mid = l + (h - l) /2;
11. **if**(nums[mid] >= target){
12. h = mid;
13. }**else**{
14. l = mid + 1;
15. }
16. }
18. **if**(nums[l] < target) **return** l+1;
20. **return** l;
21. }

## 3. 查找第最后一个no

A screenshot of a cell phone

Description automatically generated

## 4. 记忆要点

不要管是找yes/no,因为yes/no会根据谓词p来回调换，要记是找第一个还是最后一个。

共同点：while(low<high), 每次符合条件先更新high

不同点：

* 找第一个t： [no, no, no….yes(t), yes, …yes] or [yes, yes, ……no(t), no, …no]
  1. Mid是没有加1
  2. 是从左边逼近，就是hi=mid, low = mid +1;
* 找最后一个t: ： [no, no, no….no(t), yes, …yes] or [yes, yes, ……yes(t), no, …no]

1. Mid 有加1
2. 从右边逼近， 就是hi=mid-1, low = mid;

高级二分典型例题：

410 [Split Array Largest Sum](https://leetcode.com/problems/split-array-largest-sum)（找第一个t）

1011 [Capacity To Ship Packages Within D Days](https://leetcode.com/problems/capacity-to-ship-packages-within-d-days)（找第一个t）

1231 [Divide Chocolate](https://leetcode.com/problems/divide-chocolate)（找最后一个t）

More Good Binary Search Problems. Here are some similar binary search problems.

1482 [Minimum Number of Days to Make m Bouquets](https://leetcode.com/problems/minimum-number-of-days-to-make-m-bouquets/discuss/686316/javacpython-binary-search/578488)

1283 [Find the Smallest Divisor Given a Threshold](https://leetcode.com/problems/find-the-smallest-divisor-given-a-threshold/discuss/446376/javacpython-bianry-search/401806)

875 [Koko Eating Bananas](https://leetcode.com/problems/koko-eating-bananas/discuss/152324/C++JavaPython-Binary-Search)

774 [Minimize Max Distance to Gas Station](https://leetcode.com/problems/minimize-max-distance-to-gas-station/discuss/113633/Easy-and-Concise-Solution-using-Binary-Search-C++JavaPython)

# Depth-first search

## 和Tree结合 中序遍历

都是在中序遍历中，处理的步骤加点东西，典型例题：

98 [Validate Binary Search Tree](https://leetcode.com/problems/validate-binary-search-tree)

99 [Recover Binary Search Tree](https://leetcode.com/problems/recover-binary-search-tree)

## 和图结合的

[133. Clone Graph](https://leetcode.com/problems/clone-graph/) 带返回值

[200 Number of Islands](https://leetcode.com/problems/number-of-islands/)

[323. Number of Connected Components in an Undirected Graph](https://leetcode.com/problems/number-of-connected-components-in-an-undirected-graph/) 同一类型的题目，都是用dfs找连通分量

[261 Graph Valid Tree](https://leetcode.com/problems/graph-valid-tree/) 经典DFS判断是否有环，带返回值, 建立邻接矩阵

[329. Longest Increasing Path in a Matrix](https://leetcode.com/problems/longest-increasing-path-in-a-matrix/) 带memorize的的图遍历，带返回值

[679. 24 Game](https://leetcode.com/problems/24-game/) 难题，基本思路能想出来，但是实现却也很难， 带返回值

[394. Decode String](https://leetcode.com/problems/decode-string/) Medium但却是经典题，主要是对于图的定义，这道题不是要遍历全图，而是要找到最后一个节点。 可以想象成每个点下面有几种操作，沿着题目的输入走一条通路，最后的那个点就是要求的。

[494. Target Sum](https://leetcode.com/problems/target-sum/) [Medium] 属于有DP思想的带memorize的dfs

[332. Reconstruct Itinerary](https://leetcode.com/problems/reconstruct-itinerary/) [Medium] 经典DFS找一条valid路径

[490. The Maze](https://leetcode.com/problems/the-maze/) [Medium] 带返回值，四个方向图遍历问题

[785. Is Graph Bipartite?](https://leetcode.com/problems/is-graph-bipartite/) [Medium]

[802. Find Eventual Safe States](https://leetcode.com/problems/find-eventual-safe-states/) [Medium] 这两道题是经典的white-grey-black DFS染色问题，都是在图里找是否有环。一个是有向图，一个是无向图。算法讲解：

<https://segmentfault.com/a/1190000005687907>

[417. Pacific Atlantic Water Flow](https://leetcode.com/problems/pacific-atlantic-water-flow/) [Medium] 不带返回值，遍历全图。

[542. 01 Matrix](https://leetcode.com/problems/01-matrix/) [Medium] 相当于DP的dfs,运用pre\_val 本质上实现了dp，不带返回值

[968. Binary Tree Cameras](https://leetcode.com/problems/binary-tree-cameras/) [Hard] DFS + 贪心，运用状态机，自底向上遍历

## 总结

DFS需要的几点：

1. 结束条件，一般是到达四边，这个应该在开头检查而不是在for里面检查
2. 有visited 数组
3. 特殊结束条件，根据题目的要求来的
4. 是否需要在主函数对每个点进行dfs

# Breadth-first search

1. 典型例题

[314. Binary Tree Vertical Order Traversal](https://leetcode.com/problems/binary-tree-vertical-order-traversal/)