

爱彼迎面经

### 1. Alien Dictionary 3 (拓扑) LC

<https://leetcode.com/problems/alien-dictionary/>

### 2. Preference List 3 (拓扑)

/\*\*

# 你有list of list, 这些叫preference list。

# 例如 :

# [[3, 5, 7, 9],

# [2, 3, 8],

# [5, 8]]

# 然后你要根据这个输入, 输出一个总的preference list

\* 这这一题应该就是 :

\* [2, 3, 5, 8, 7, 9]

\*/

### 3. Boggle Game 3 (DFS+Trie)

<https://www.geeksforgeeks.org/boggle-find-possible-words-board-characters/>

<https://leetcode.com/problems/word-search-ii/description/>

### 4. Flight Ticket 3 (DFS)

<https://leetcode.com/problems/cheapest-flights-within-k-stops/description/>

给定很多航班信息, 至多k stop, 找最便宜路线。给很多tuple <depart city, dest city, cost> 代表flight, 给定 city A, city B, maxStops, 求最小cost的path。

显然如果没有maxStops的限制, 这就是一个经典的最短路径问题, 可以使用Dijkstra's algorithm计算cityA 到 cityB的最短路径 (花费表示路径)。

接下来考虑最多在只能经过maxStops个点, 这就是一个带有点限制的最短路径问题, 我们把Dijkstra的点i拆分成maxStop个点

[i][0], [i][1]。。。表示经过0个点到i, 经过1个点到i, 经过2个点到i,

重新构图

[k][i - 1] -> [i][i] 显然之前经过i-1个点, 之后就是经过i个点, 是否能连边, 取决于第二位是否相邻。

那么重新构图以后显然就是求 [cityA][0] -> [cityB][0].....[cityB][maxStops]的最短路中取最小值。

### 5. Flood Fill 3 (BFS/DFS)

<https://leetcode.com/problems/flood-fill/description/>

6. Hilbert Curve 3 (DFS)

<http://blog.csdn.net/yaoxiaochuang/article/details/50695142>

```
public static int hilbert_curve(int x, int y, int iter){
    //base case
    if(iter == 0)
        return 1;
    //harfLen is used to determine which part the point belong to,for i
    int harfLen = (1 << (iter - 1));
    int harfNum = (1 << (2 * iter - 2));

    if(x >= harfLen && y >= harfLen)
        return 2 * harfNum + hilbert_curve(x - harfLen, y - harfLen, iter - 1);
    else if (x < harfLen && y >= harfLen)
        return harfNum + hilbert_curve(x, y - harfLen, iter - 1);
    else if (x < harfLen && y < harfLen)
        return hilbert_curve(y, x, iter - 1);
    else
        return 3 * harfNum + hilbert_curve(harfLen - 1 - y, 2 * harfLen - 1 - x, iter - 1);
}
```

7. Menu Combination Sum 3 (DP/DFS)

变种<https://leetcode.com/problems/combination-sum/>

8. Wizards Distance 3 (BFS)

There are 10 wizards, 0-9, you are given a list that each entry is a list of wizards known by wizard. Define the cost between wizards and wizard as square of different of i and j. To find the min cost between 0 and 9.

9. House Robber With Index 3 (DP)

<https://leetcode.com/problems/house-robber/description/>

<https://leetcode.com/problems/house-robber-ii/description/>

<https://leetcode.com/problems/house-robber-iii/description/>

10. Maximum Square

3 (DP)

LC

<https://leetcode.com/problems/maximal-square/description/>

11. Regular Expression 3 (DP) LC  
<https://leetcode.com/problems/regular-expression-matching/description/>

12. String Pyramids Transition Matrix 3 (DP)  
<https://leetcode.com/problems/pyramid-transition-matrix/description/>

design 题 [https://github.com/jxr041100/system\\_design](https://github.com/jxr041100/system_design)

13. Circular Buffer 3 (Design)  
<http://www.cnblogs.com/shanyou/archive/2013/02/04/2891300.html>  
<http://blog.csdn.net/sidihuo/article/details/44928843>

14. File System 3 (Design)  
<https://leetcode.com/problems/design-in-memory-file-system/description/>

15. Puzzle 3 (Design+BFS)

16. Two D Iterator 3 (Design)  
[https://github.com/jxr041100/system\\_design/blob/master/Airbnb:%20D%20Iterator%20with%20remove\(\)](https://github.com/jxr041100/system_design/blob/master/Airbnb:%20D%20Iterator%20with%20remove())

17. CSV Parser 2 (模拟)

18. Display Page List 2 (模拟)  
[https://github.com/jxr041100/system\\_design/blob/master/Airbnb:%20Page%20Display](https://github.com/jxr041100/system_design/blob/master/Airbnb:%20Page%20Display)

19. IP to Cidr 2 (模拟+bit)  
CIDR = Classless Inter-Domain Routing  
Concept: <http://uule.iteye.com/blog/2102484>  
<https://stackoverflow.com/questions/5020317/in-java-given-an-ip-address-range-return-the-minimum-list-of-cidr-blocks-that>  
<https://leetcode.com/problems/ip-to-cidr/description/>

20. Text Justification 2 (模拟) LC

21.Travel BuddyList 2 (模拟)

<https://leetcode.com/problems/pour-water/description/>

<https://leetcode.com/problems/meeting-rooms/description/>  
<https://leetcode.com/problems/meeting-rooms-ii/description/>

## 25. Number Of Connected Components 3 (UnionFind) LC

<https://leetcode.com/problems/number-of-connected-components-in-an-undirected-graph/description/>

27. Palindrome Pairs 3 (Trie) LC

<https://leetcode.com/problems/palindrome-pairs/description/>

<https://leetcode.com/problems/basic-calculator/description/>  
<https://leetcode.com/problems/basic-calculator-ii/description/>  
<https://leetcode.com/problems/basic-calculator-iii/description/>  
<https://leetcode.com/problems/basic-calculator-iv/description/>

[http://blog.csdn.net/java2000\\_net/article/details/2397982](http://blog.csdn.net/java2000_net/article/details/2397982)

<https://leetcode.com/problems/mini-parser/>

### 31. Rounded Number 3