LeetCode 723. Candy Crush

- **Difficulty:Medium**
- Total Accepted:3.4K
- Total Submissions:12.7K

This question is about implementing a basic elimination algorithm for Candy Crush.

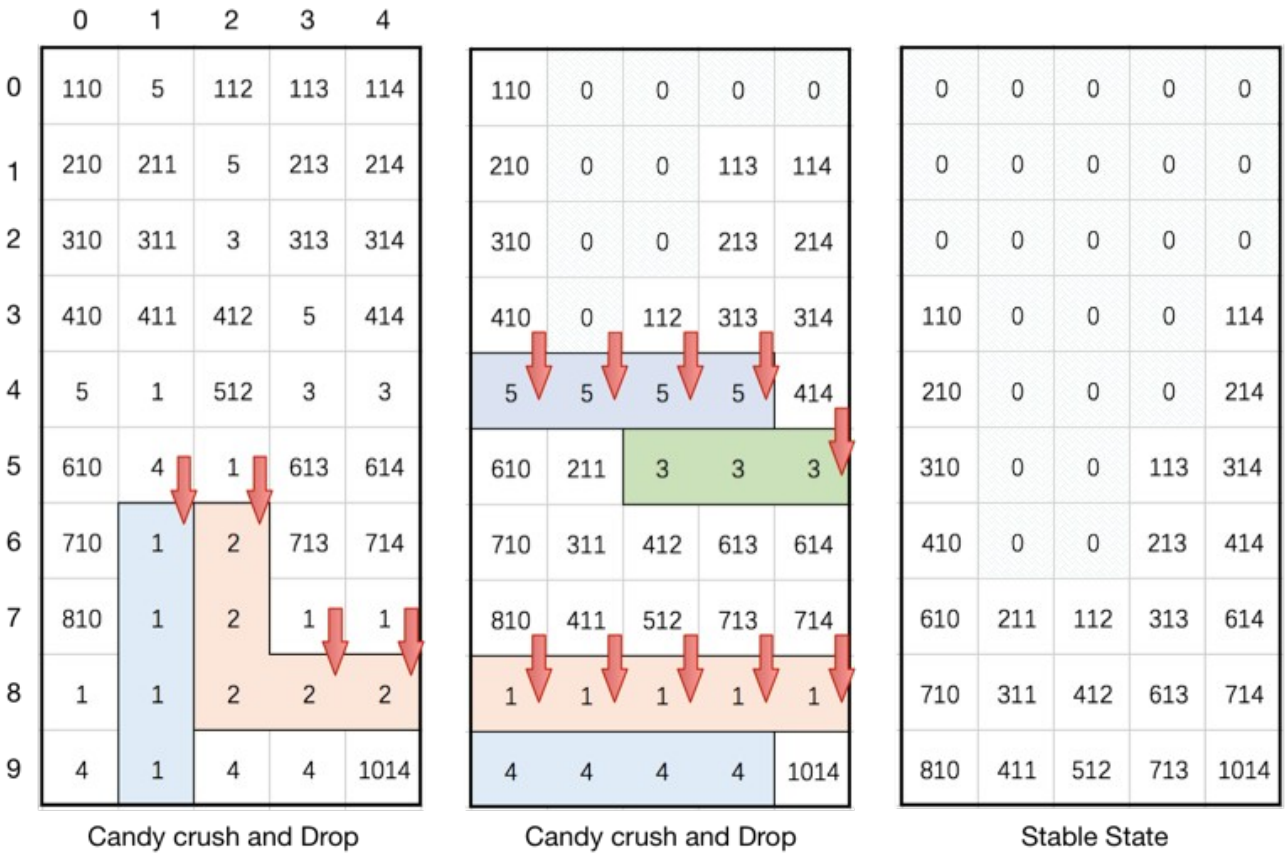
Given a 2D integer array `board` representing the grid of candy, different positive integers `board[i][j]` represent different types of candies. A value of `board[i][j] = 0` represents that the cell at position `(i, j)` is empty. The given board represents the state of the game following the player's move. Now, you need to restore the board to a *stable state* by crushing candies according to the following rules:

1. If three or more candies of the same type are adjacent vertically or horizontally, "crush" them all at the same time - these positions become empty.
2. After crushing all candies simultaneously, if an empty space on the board has candies on top of itself, then these candies will drop until they hit a candy or bottom at the same time. (No new candies will drop outside the top boundary.)
3. After the above steps, there may exist more candies that can be crushed. If so, you need to repeat the above steps.
4. If there does not exist more candies that can be crushed (ie. the board is *stable*), then return the current board.

You need to perform the above rules until the board becomes stable, then return the current board.

Example 1:

```
Input:
board =
[[110,5,112,113,114],[210,211,5,213,214],[310,311,3,313,314],
[410,411,412,5,414],[5,1,512,3,3],[610,4,1,613,614],[710,1,2,713,714],
[810,1,2,1,1],[1,2,1,2,2],[4,1,4,4,1014]]
Output:
[[0,0,0,0,0],[0,0,0,0,0],[0,0,0,0,0],[110,0,0,0,114],[210,0,0,0,214],
[310,0,0,113,314],[410,0,0,213,414],[610,211,112,313,614],[710,311,412,613,714],
[810,411,512,713,1014]]
Explanation:
```



Note:

1. The length of `board` will be in the range [3, 50].
2. The length of `board[i]` will be in the range [3, 50].

3. Each `board[i][j]` will initially start as an integer in the range [1, 2000].

题目大意：

模拟实现游戏“糖果粉碎传奇”“开心消消乐“

给定二维数组 board，每行、列中 3 个或以上连续相同的数字都可以被消去（变为 0 表示空位），消去后位于上方的数字会填充空位。

重复直到没有更多的数字可以被消去

```
#include<stdio.h>
#include<iostream>
#include<vector>
#include<algorithm>
#include<stdlib.h>
#include<string.h>
using namespace std;

void drop(vector<vector<int>>& board, vector<vector<int>>& visit,
int x, int y)
{
     int m = board.size();
     int n = board[0].size();
     if(board[x][y]==0) return;
     int cnt = 0;
     for(int i=x;i<m;i++)
     {
          if(board[x][y]!=board[i][y]) break;
          cnt++;
     }
     if(cnt>=3)
     {
          for(int i=0;i<cnt;i++) visit[i+x][y]=1;
     }
     cnt=0;
     for(int i=y;i<n;i++)
     {
          if(board[x][i]!=board[x][y]) break;
          cnt++;
     }
     if(cnt>=3)
     {
          for(int i=0;i<cnt;++i) visit[x][i+y]=1;
     }
}
```

```
bool check(vector<vector<int>>& board, vector<vector<int>>& visit)
{
     int m = board.size();
     int n = board[0].size();
     for(int i=0;i<m;i++)
          for(int j=0;j<n;j++) visit[i][j]=0;
     for(int i=0;i<m;i++)
          for(int j=0;j<n;j++)
               drop(board,visit,i,j);

     int z[55]; memset(z, 0, sizeof(z));
     bool flag = false;
     for(int i=0;i<m;i++)
          for(int j=0;j<n;j++)
          {
               if(visit[i][j]==1)
               {
                    flag=true;
                    for(int k=i;k>0;k--)
                         board[k][j]=board[k-1][j];
                    board[z[j]][j]=0;
                    z[j]++;
               }
          }
     return flag;
}

vector<vector<int>> candyCrush(vector<vector<int>>& board) {
     int m = board.size();
     int n = board[0].size();
     vector<vector<int>> visit(m,vector<int>(n,0));
     int counter = 0;
     while(check(board,visit))
     {
          counter++;
          printf("iteration %d steps\n",counter);
     }
     vector<vector<int>> ret(m,vector<int>(n,0));
     for(int i=0;i<m;i++)
          for(int j=0;j<n;j++)
          ret[i][j]=board[i][j];
     return ret;
}
int main(int argc, char *argv[])
{
     vector<vector<int>> test = {{110,5,112,113,114},
{210,211,5,213,214},{310,311,3,313,314},
          {410,411,412,5,414},{5,1,512,3,3},{610,4,1,613,614},
          {710,1,2,713,714},{810,1,2,1,1},{1,1,2,2,2},
{4,1,4,4,1014}};
     vector<vector<int>> ans = candyCrush(test);
     for(int i=0;i<(int)ans.size();i++)
```

```
    {
        for(int j=0;j<(int)ans[0].size();++j)
        {
            printf("%d ",ans[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```