The Curious Case Of SoMayGen

We have N monkeys, P pizzas (circular), and K knives.
It was clearly mentioned that one knife can be used for performing one cut.

In order to find the minimum number of monkeys which will revolt, we need to find the maximum number of slices which can be generated from the P pizzas, given we perform straight cuts.

This is when we perform all cuts on the same pizza, which generate (K*(K+1))/2 + 1 slices.
http://mathworld.wolfram.com/CircleDivisionbyLines.html

Since monkeys eat pizza slices irrespective of shape and size, we have more P - 1, slices remaining.

Hence the total number of slices is, (K*(K+1))/2 + 1 + P - 1

One SIMPLE test case which many submissions were failing at is when, we have non-zero monkeys and knives but zero pizzas, that would mean, all monkeys will revolt.

Save The World

So the problem's understanding goes like:
there is a grid of at max 51*51 as X and Y can go up to 50(0 indexed) only and the holes are some points in the grid.

Let Zero(0) be simple ground and one(1) represent a hole in the ground.

So the given test case is like:

1010
0111
1110
0010

Now, Gandalf wants to block the holes, so as to prevent us from orcs' attack. The optimal answer for the above problem is 4. Now let us see how it is achieved?

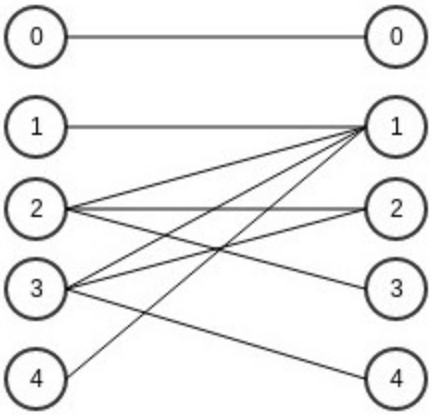
First of all each hole can be covered with either a vertical plank or a horizontal plank or both. Now Let us look at a catch in the problem, in order to solve the problem optimally if a hole is covered with a horizontal plank, all of it's continuous horizontal neighbors should be covered by it. So you start numbering the planks from 0 and increment the number as soon as you have to introduce a new plank. You repeat the similar process for numbering the vertical planks.

Now for each hole you have got a vertical plank number and a horizontal plank number. Now the important part comes, The holes must not be such that both the horizontal and vertical plank is dedicated it only. So we can have a Bipartite graph of horizontal plank number and vertical plank number and make an edge for each hole from it's horizontal plank number to it's vertical plank number. If you look closely, the problem reduces directly to the minimum vertex cover problem where each plank numbers are vertices and you need to find the minimum covering of the bipartite graph.

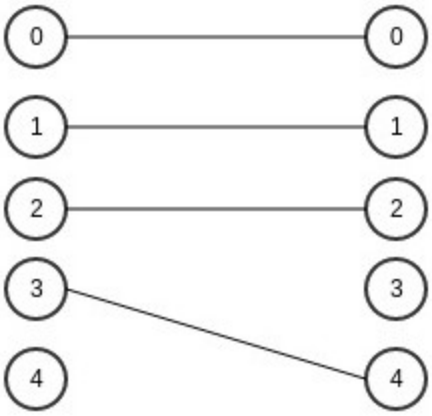
Now let us run the algorithm on the above problem:
for hole (0,0) HorizontalPlankNumber = 0, VerticalPlankNumber = 0
for hole (0,2) HorizontalPlankNumber = 1, VerticalPlankNumber = 1
for hole (1,1) HorizontalPlankNumber = 2, VerticalPlankNumber = 2
for hole (1,2) HorizontalPlankNumber = 2, VerticalPlankNumber = 1
for hole (1,3) HorizontalPlankNumber = 2, VerticalPlankNumber = 3

for hole (2,0) HorizontalPlankNumber = 3, VerticalPlankNumber = 4
for hole (2,1) HorizontalPlankNumber = 3, VerticalPlankNumber = 2
for hole (2,2) HorizontalPlankNumber = 3, VerticalPlankNumber = 1
for hole (3,2) HorizontalPlankNumber = 4, VerticalPlankNumber = 1

The Bipartite Graph looks like:



One of the maximal matching of the above graph or the minimum vertex cover (Konig's Theorem) of the above graph is:



So the maximal matching of the above graph is 4, which is our answer.

Sharkland

Type :- greedy

Mr Raj can buy sharks on some days and sell them at most once in n days. Best way to maximise profit is to buy as many sharks on a day with minimum buying cost and sell all of them on a day with maximum selling cost. Obviuosly the selling day > buying day. Note :- it is given buying cost = selling cost for a given day.
For a particular day i, we want to find the day with maximum selling cost in all days > i, This can be found in O(1) after preprocessing the maximum in range(i, n). [i.e. dp[i] will store the day with maximum selling cost that comes after i]
After that find the maximum profit for all i in [1,n)

Complexity - O(n)

Force

Key Points:

1. We first have to notice that the if the sum of the strengths of all the Jedi's is strictly less than K, then Yoda's process will never complete.
2. If the number of remaining iterations are greater than N, then the strengths of all the jedi's have to be reduced by 1.
3. We can apply the step number 2 until the number of remaining iterations fall below N and we have to keep track of what the status of the queue will be when this happens.
4. If the number of remaining iterations are less than or equal to N, then we can simulate the remaining iterations as the maximum value of N <= 100000. So, it will run in time.

Class Chaos

We can describe the current state in the classroom with a bitmask, where the value of each bit tells
whether a child was hit by an even or odd number of chalks during the previous lecture. There are 2^N of these states, or about 10^6 for N=20. With a number of lectures larger than this, a state is
guaranteed to repeat and, once it does, it will keep looping through the same cycle. We can calculate the
number of throws in one iteration of the cycle in O(N·2^N) and, with careful implementation, skip most
of the iterations through the cycle.