**Problem :** [**https://codingcompetitions.withgoogle.com/kickstart/round/0000000000435bae/0000000000888764**](https://codingcompetitions.withgoogle.com/kickstart/round/0000000000435bae/0000000000888764)

**Approach :**

-> Can start from any node and collect those magic points.

-> If by collecting those magic points , no of points = k, return 1, as this indicated 1 path.

-> We go to dfs of a node only if it can be visited. When we start that node’s dfs, we mark it as **visited in the visited bitmask**, and in another bitmask called **reachable,** we mark all the nodes which are reachable from the current node.

-> So now for all the nodes we check , whether it’s **not visited(bit=0 in visited mask)** and it’s **reachable by one of the nodes in the path(the reachable mask shows it as 1),** and the magic points it required to break the shield is less or equal to what we have currently,

if all 3 conditions met, we go and visit this node, as this may contribute in a unique permutation.

**Answer for current visited mask = sum of all the valid paths we got by doing dfs on valid children of current node.**

**=> Offcoure if dp[visitedmask] is calculated once, you don’t do that again, as it will give same number of permutations.**

**So time = O(2^15 permutations \* (15) transition time)** which easily pass 1 sec.