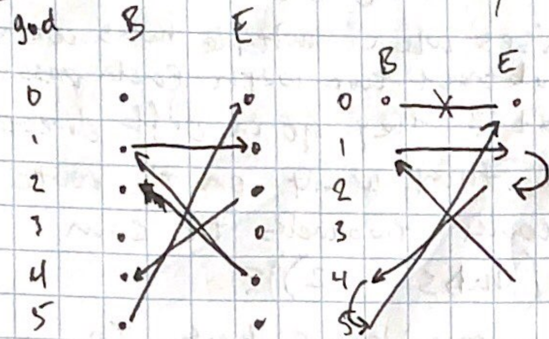
23 Nov 2020 XC5 - 12 xjc 1636

1. Path in a bipartite graph?
2. ~~tr~~ tree dp? might be disconnected
3. Sparse table then ~~a~~ monotonic pass?
   ! need long long
   Could also binary search and do grouping...
   bin search is probably easier



It's a directed bipartite graph
(almost, bc she can give more valuable pies)
and from any point, ask the shortest path to zero... dp I guess?

Can't just add edges to simulate D...

naive D simulation would be O(ND) which is TLE
but you could sort pies by start and then it would be O(N) amortized?

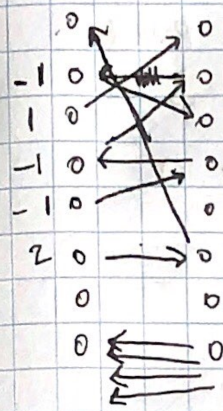
isn't it just a bfs then? for each taste on each side, store a list of edges leaving from there?

...

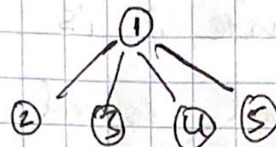
map < tastiness, list <eid> > eg[2];
vis[eid], ans[eid], taste[2][eid]

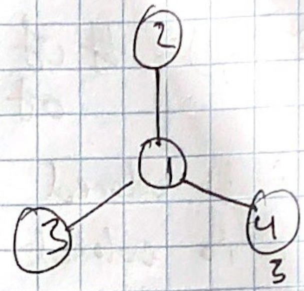
queue.push(0)
side = 0 ← need to run for ~~each~~ both sides
while queue:
for ~~cur sto~~
it = lower_bound(taste[side][cur]);
taste[!side][~~cur~~ it] <= taste[!side][cur] + D;
++it:
ans[it] = ans[cur] + 1 ← minify?
queue.push it
~~i~~ map.erase(cur)

cont #2



1	2
2	0
3	1
4	3
5	5
3	3
5	0
7	7
7	7

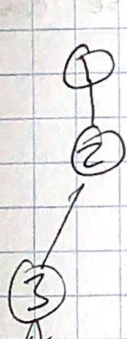


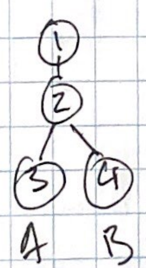


dp[i][j] = # of ways to paint subtree i if i should be painted j

$$dp[i][j] \begin{cases} \sum \prod dp[c(i)][!j] & \text{if } i \text{ not painted or paint}[i] = j \\ 0 & \text{if paint}[i] \text{ \&\& paint}[i] \; != j \end{cases}$$

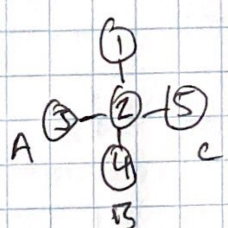
What if a child forces a color?



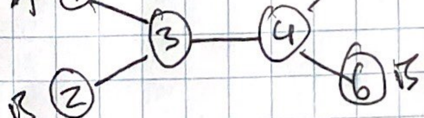


← dp[1][A] = 0

I think it still works as long as it doesn't affect farther than one



here, two will always break



this is also impossible... both 3 and 4 are forced into being color C

maybe casework:

if 1 child colored: return zero on that color

2: set own color

3: panic

maybe it should be dp[i][j] = parent of i is j cont #2 try more cases, I guess

dp[i][j] = # of ways in subtree if parent of node i is color j

if i is colored and colored j: ret 0

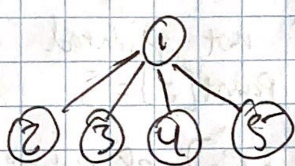
if i is colored not j:

ret $\prod$ dp[c[i]][color[i]]

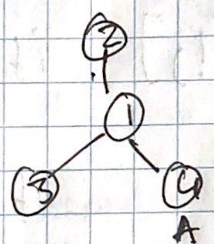
else, ret $\sum$ for each color x != j

$\prod$ dp[c[i]][x]

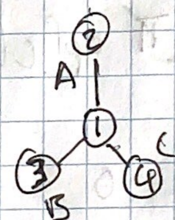
now lets try some test cases



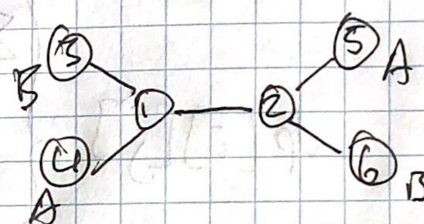
3 x 16



= 8 ✓

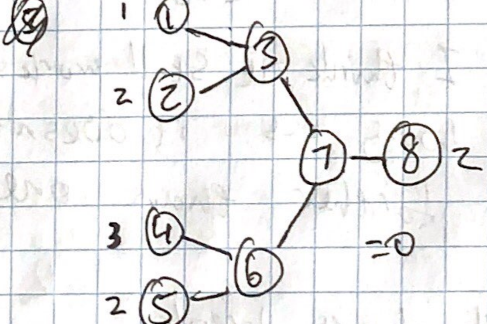


= 0 ✓



= 0 ✓

I mean this equation seems to work with these cases...



= 0