

5970. Maximum Employees to Be Invited to a Meeting

[My Submissions \(/contest/weekly-contest-274/problems/maximum-employees-to-be-invited-to-a-meeting/submissions/\)](/contest/weekly-contest-274/problems/maximum-employees-to-be-invited-to-a-meeting/submissions/)

[Back to Contest \(/contest/weekly-contest-274/\)](/contest/weekly-contest-274/)

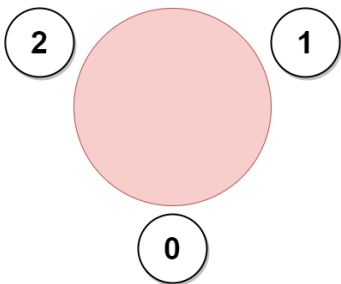
A company is organizing a meeting and has a list of n employees, waiting to be invited. They have arranged for a large **circular** table, capable of seating **any number** of employees.

The employees are numbered from 0 to $n - 1$. Each employee has a **favorite** person and they will attend the meeting **only if** they can sit next to their favorite person at the table. The favorite person of an employee is **not** themselves.

Given a **0-indexed** integer array `favorite`, where `favorite[i]` denotes the favorite person of the i^{th} employee, return the **maximum number of employees** that can be invited to the meeting.

User Accepted:	0
User Tried:	0
Total Accepted:	0
Total Submissions:	0
Difficulty:	Hard

Example 1:



Input: `favorite = [2,2,1,2]`

Output: 3

Explanation:

The above figure shows how the company can invite employees 0, 1, and 2, and seat them at the round table. All employees cannot be invited because employee 2 cannot sit beside employees 0, 1, and 3, simultaneously. Note that the company can also invite employees 1, 2, and 3, and give them their desired seats. The maximum number of employees that can be invited to the meeting is 3.

Example 2:

Input: `favorite = [1,2,0]`

Output: 3

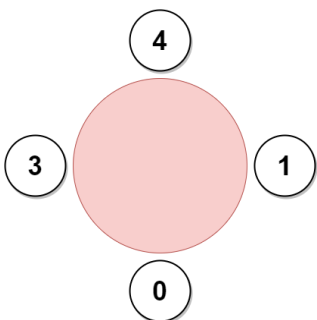
Explanation:

Each employee is the favorite person of at least one other employee, and the only way the company can invite them is if they all sit at the table. The seating arrangement will be the same as that in the figure given in example 1:

- Employee 0 will sit between employees 2 and 1.
- Employee 1 will sit between employees 0 and 2.
- Employee 2 will sit between employees 1 and 0.

The maximum number of employees that can be invited to the meeting is 3.

Example 3:



Input: favorite = [3,0,1,4,1]

Output: 4

Explanation:

The above figure shows how the company will invite employees 0, 1, 3, and 4, and seat them at the round table. Employee 2 cannot be invited because the two spots next to their favorite employee 0 are taken. So the company leaves them out of the meeting. The maximum number of employees that can be invited to the meeting is 4.

Constraints:

- $n == \text{favorite.length}$
- $2 \leq n \leq 10^5$
- $0 \leq \text{favorite}[i] \leq n - 1$
- $\text{favorite}[i] \neq i$

JavaScript



```

1  const initializeGraph = (n) => { let g = []; for (let i = 0; i < n; i++) { g.push([]); } return g; };
2  const mx = Math.max;
3
4  let g;
5  const maximumInvitations = (a) => {
6    let n = a.length, res = 0, visit = new Set();
7    g = initializeGraph(n);
8    for (let i = 0; i < n; i++) g[a[i]].push(i);
9    for (let i = 0; i < n; i++) {
10     res += i < a[i] || a[a[i]] !== i ? 0 : dfs(i, a[i]) + dfs(a[i], i);
11   }
12   for (let i = 0; i < n; i++) {
13     let j = i;
14     if (!visit.has(i)) {
15       let m = new Map();
16       for (let k = 0; !visit.has(j); k++) {
17         visit.add(j);
18         m.set(j, k);
19         j = a[j];
20       }
21       let tmp = m.size - m.get(j) || 0;
22       res = mx(res, tmp);
23     }
24   }
25   return res;
26 };
27
28 const dfs = (i, j) => {
29   let max = 0;
30   for (const child of g[i]) {
31     max = mx(max, child === j ? 0 : dfs(child, j));
32   }
33   return 1 + max;
34 };

```

☐ Custom Testcase

Use Example Testcases

Run

Submit

Submission Result: **Accepted** (/submissions/detail/611234663/) ?

More Details > (/submissions/detail/611234663/)

Share your acceptance!