# Problem D
# Count DFS Graph

You are currently researching a graph traversal algorithm called the Depth First Search (DFS). Starting with an empty list `A`, the following pseudocode will fill the list `A` with the visitation order of a DFS algorithm.

```
DFS(v):
    append v to A
    for each u neighbour of v in ascending node number:
        if u is not in A:
            DFS(u)
```

After running `DFS(1)` from the pseudocode above, you now have a list $A$ containing a permutation of integers from $1$ to $N$. You now wonder how many different simple undirected graphs with $N$ nodes there are that yield the list $A$ that you have. Count the number, modulo $998\,244\,353$.

A graph is simple when there are no self-loops and there is at most one edge connecting each pair of nodes. Two graphs are considered different if there is an edge connecting a pair of nodes in one graph but not the other.

### Input

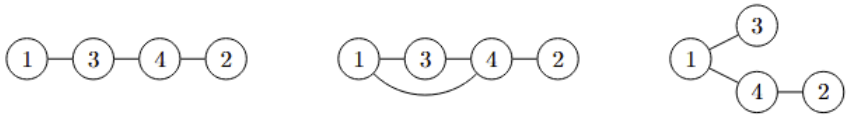
The first line contains an integer $N$ ($2 \leq N \leq 300$). The second line contains a permutation of the first $N$ positive integers, representing the list $A$. The first element of $A$ is guaranteed to be $1$.

### Output

A single integer representing the number of different graphs, whose DFS order gives you the list $A$, modulo $998\,244\,353$.

| Sample Input 1 | Sample Output 1 |
|---|---|
| 4<br>1 3 4 2 | 3 |

*Explanation of Sample 1:* The following illustrates all the graphs with the given DFS order.




| Sample Input 2 | Sample Output 2 |
|---|---|
| 10<br>1 2 3 4 5 6 7 8 9 10 | 515546413 |

*This page is intentionally left blank.*