**Towers**

**Link submit:** <https://codeforces.com/problemset/problem/37/A>

**Solution:**

|  |  |
| --- | --- |
| C++ | <https://ideone.com/zwiAj1> |
| Java | <https://ideone.com/fqoxGU> |
| Python | <https://ideone.com/lNsc2D> |

**Tóm tắt đề:**

Vasya có N thanh gỗ với các chiều dài khác nhau. Hai thanh gỗ cùng chiều dài có thể chồng lên nhau để tạo thành một tòa tháp.

Bạn hãy giúp Vasya xây tháp sao cho số lượng tháp được xây là nhỏ nhất. Biết rằng mọi thanh gỗ đều phải được sử dụng.

**Input:**

Dòng đầu tiên chứa một số nguyên *N* (1 ≤ N ≤ 1.000) – số lượng thanh gỗ Vasya có.

Dòng tiếp theo gồm N số nguyên *li* (li ≤ 1.000) đại diện cho chiều dài của thanh gỗ thứ i.

**Output:**

In ra trên cùng một dòng hai số nguyên lần lượt là chiều cao của tòa tháp cao nhất và số lượng tòa tháp ít nhất được xây.

**Ví dụ:**

|  |  |
| --- | --- |
| 3 1 2 3 | 1 3 |

|  |  |
| --- | --- |
| 4 6 5 6 7 | 2 3 |

**Giải thích ví dụ:**

***Ví dụ 1:*** Có 3 thanh gỗ với chiều dài phân biệt là {1, 2, 3}. Do đó ta có thể xây được 3 tòa tháp, mỗi tòa tháp đều có độ cao bằng 1.

***Ví dụ 2:*** Có 4 thanh gỗ với chiều dài là {5, 6, 6, 7}. Ta có thể ghép 2 thanh gỗ 6 với nhau tạo thành một tòa tháp có độ cao là 2. Các thanh gỗ còn lại có chiều cao 5 và 7, mỗi thanh tạo thành một tòa tháp phân biệt với độ cao là 1. Do đó, ta xây được tổng cộng 3 tòa tháp và tòa tháp cao nhất có độ cao là 2.

**Hướng dẫn giải:**

Nhận xét:

* Số tòa tháp chính bằng số chiều dài phân biệt và chiều cao của tòa tháp cao nhất bằng số lượng lớn nhất các thanh gỗ cùng chiều dài.
* Để đếm số chiều dài phân biệt, ta có thể sắp xếp lại mảng theo thứ tự tăng dần hoặc giảm dần. So sánh lần lượt từng cặp phần tử kề nhau và tăng biến đếm số phần tử phân biệt nếu hai phần tử đó khác nhau.
* Đồng thời, với mỗi lần ta duyệt đến một phần tử phân biệt mới, ta bắt đầu đếm số lần xuất hiện của phần tử ấy và liên tục cập nhật với số lần xuất hiện lớn nhất.

Tóm lại, ta có cách làm như sau:

* Bước 1: Đưa thông tin chiều dài các thanh gỗ vào mảng.
* Bước 2: Sắp xếp mảng tăng dần.
* Bước 3: Lấy thanh gỗ đầu tiên làm gốc, ta sử dụng một biến đếm số lượng thanh gỗ phân biệt, khởi tạo bằng 1 (tức xem phần tử ở đầu mảng là phần tử phân biệt đầu tiên). Biến đếm số lượng thanh gỗ có cùng chiều dài với thanh gỗ đang xét, khởi tạo bằng 1.
* Bước 4: Lần lượt duyệt qua các thanh gỗ có chiều dài từ bé đến lớn, không tính thanh đầu tiên:
  + Nếu thanh gỗ này có cùng chiều dài với thanh trước đó thì ta tăng biến đếm số lượng thanh gỗ có cùng chiều dài lên 1. Đồng thời, so sánh số lượng thanh gỗ có cùng chiều dài hiện tại với số lượng lớn nhất và liên tục cập nhật lại.
  + Ngược lại, tăng biến đếm số lượng phần tử phân biệt lên 1 và cập nhật lại số lượng thanh gỗ có cùng chiều dài với thanh gỗ đang xét bằng 1.
* Bước 5: In kết quả.

**Độ phức tạp:** **O(NlogN)** với N là số lượng thanh gỗ.