



## Kurt Problemi

Aşağıdaki problemi Java, C#, Kotlin ya da Javascript ile geliştiriniz.

Her ögenin bir kurt türü ID'sini temsil ettiği bir dizi kurt gözlemi içerisinde, en sık tespit edilen türün en düşük ID'sini belirlemeye çalışalım. Maksimum sayıya erişen 1'den fazla tür tespit edildiyse, ID'si en küçük olanı döndürmeniz beklenmektedir.

### Örnek

*dizi = [1,1,2,2,3]*

Örnek dizide her birinden 2 defa tespit edilen 2 tür (1 ve 2) ve 1 defa tespit edilen 1 tür (3) bulunuyor. Bunlar arasında en fazla görülen en düşük ID **1**'dir.

### Fonksiyon Tanımı

Bu problem aşağıdaki parametrelere sahiptir:

int dizi[n]: tespit edilen kurt türleri

### Döndürülmesi Beklenen

int: En çok tespit edilen en düşük ID değeri.

### Veri Giriş Formatı

İlk satır, dizi boyutunda bir tamsayı (n) içerir.

İkinci satır, diziyi, her biri tespit edilen kurtların bir tür numarası olan boşlukla ayrılmış n tamsayı olarak tanımlar.

### Kısıtlamalar

$$5 \leq n \leq 2 \times 10^5$$

Her türün 1,2,3,4 veya 5 olduğu garanti edilir.

### Örnek Senaryo

11

1 2345432134

### Beklenen Çıktı

3

Geliştirdiğiniz kodu GitHub profiliniz üzerinden bizimle paylaşmanızı rica ederiz.



## Java Kod Şablonu

```
import java.io.*;
import java.math.*;
import java.security.*;
import java.text.*;
import java.util.*;
import java.util.concurrent.*;
import java.util.function.*;
import java.util.regex.*;
import java.util.stream.*;
import static java.util.stream.Collectors.joining;
import static java.util.stream.Collectors.toList;

class Result {

    public static int uniqueWolfs(List<Integer> arr) {

        // Gelistirmeyi bu alana yapiniz

    }

}

public class Solution {

    public static void main(String[] args) throws IOException {

        BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(System.in));

        BufferedWriter bufferedWriter = new BufferedWriter(new
        FileWriter(System.getenv("OUTPUT_PATH")));

        int arrCount = Integer.parseInt(bufferedReader.readLine().trim());

        List<Integer> arr = Stream.of(bufferedReader.readLine().replaceAll("\\s+$", "").split("
")).map(Integer::parseInt).collect(toList());

        int result = Result.uniqueWolfs(arr);

        bufferedWriter.write(String.valueOf(result));
        bufferedWriter.newLine();
        bufferedReader.close();
        bufferedWriter.close();

    }

}
```