# ForkJoin Practice

## a) lessThan7

```java
public static int lessThan7(int[] arr)
Returns the number of elements in arr that are less than 7.
For example, if arr is [21, 7, 6, 8, 17, 1], then lessThan7(arr) == 2.
```

Your code must have \(O(n)\) work, \(O(\log n)\) span, where \(n\) is the length of arr

## b) parity

```java
public static boolean parity(int[] arr)
Returns true if there are even number of even numbers and false otherwise.
For example, if arr is [1, 7, 4, 3, 6], then parity(arr) == true.
But, if arr is [6, 5, 4, 3, 2, 1], parity(arr) == false.
```

Your code must have \(O(n)\) work, \(O(\log n)\) span, where \(n\) is the length of arr

## c) countStrs

```java
public static int countStrs(String[] arr, String str)
Returns the number of elements in arr that equal str
For example, if arr is ["h", "ee", "1111", "1111", "oo", "1111"], then countStrs(arr, "1111") == 3 and countStrs(arr, "h") == 1.
```

Your code must have \(O(n)\) work, \(O(\log n)\) span, where \(n\) is the length of arr

## d) secondSmallest

```java
public static int secondSmallest(int[] arr)
Returns the second smallest unique element of arr. Assume arr contains at least two unique elements.
For example, if arr is [1, 7, 4, 3, 6], then secondSmallest(arr) == 3.
But, if arr is [6, 1, 4, 3, 5, 2, 1], secondSmallest(arr) == 2.
```

Your code must have \(O(n)\) work, \(O(\log n)\) span, where \(n\) is the length of arr

## e) powmod

```java
public static void powmod(int[] arr, int p, int m)
Replaces every element of arr with arr[i]^p mod m.
For example, if arr is [1, 7, 4, 3, 6], then powmod(arr, 2, 5) would result in arr = [1, 4, 1, 4, 1].
```

Your code must have \(O(n)\) work, \(O(\log n)\) span, where \(n\) is the length of arr