

位元操作於 Java

位元操作使用位元運算子來高效解決問題。讓我們探索找到單一數字和檢查一個數字是否為 2 的幂。

1. 單一數字：XOR 魔法

在一個數組中找到唯一不重複的數字，其中其他所有數字都出現兩次，使用 XOR。

Java 實現

```
public class SingleNumber {  
    public static int singleNumber(int[] nums) {  
        int result = 0;  
        for (int num : nums) result ^= num;  
        return result;  
    }  
  
    public static void main(String[] args) {  
        int[] nums = {2, 2, 1, 1, 4};  
        System.out.println("Single number: " + singleNumber(nums));  
    }  
}
```

輸出：Single number: 4

2. 2 的幂：位元檢查

如果一個數字在其二進制表示中有且僅有一個 1 位元，則該數字是 2 的幂。

Java 實現

```
public class PowerOfTwo {  
    public static boolean isPowerOfTwo(int n) {  
        return n > 0 && (n & (n - 1)) == 0;  
    }  
  
    public static void main(String[] args) {  
        int n = 16;  
        System.out.println(n + " is power of 2: " + isPowerOfTwo(n));  
    }  
}
```

```
        n = 18;

        System.out.println(n + " is power of 2: " + isPowerOfTwo(n));
    }
}
```

輸出:

```
16 is power of 2: true
18 is power of 2: false
```

結論跨博客

- **排序**：快速排序、合併排序和氣泡排序涵蓋速度、穩定性和簡單性。
- **字串**：KMP、字典樹和 Rabin-Karp 處理模式匹配和前綴搜索。
- **搜索**：二分搜索和線性搜索提供對數和線性解決方案。
- **圖表**：DFS、BFS、Dijkstra's 和 Kruskal's 解決遍歷、路徑和最小生成樹。
- **位元操作**：XOR 和位元檢查使用最少的操作解決棘手的問題。

每個類別都有其優勢——選擇適合您問題的工具並嘗試這些實現！如果您想要更深入的內容或任何類別的其他算法，請告訴我。