

2808.使循环数组所有元素相等的最少秒数

题目大意

在一个环上有若干格子，每一回合可以让任意多格子相邻格子中填的数字变为中间格子填的数字，问至少需要多少回合可以让所有格子中的数字变的一样。

题目分析

感性理解这个题目，如果最终格子里的数字全都变成 x ，相当于初始所有格子里为 x 的作为源，实时地在向外扩散。对于两个“相邻”的源（两个初始为 x 的格子之间没有其他的 x ），若他们之间有 k 个格子，填满这 k 个格子所需回合数为 $\lceil \frac{k}{2} \rceil$ 。因此，最终全填 x 的话，所需回合数即为所有相邻源的距离通过除2上取整的最大值。

对于这个问题，统计所有出现过的颜色的答案，取最小值，即为最优解。

下面给出代码，复杂度为 $O(n)$ ，由于数字范围比较大，为了节约时间，可以考虑开unordered_map记录数据。下面是代码示例。

```
unordered_map<int, int> last; // 最后一次出现的位置
unordered_map<int, int> first; // 第一次出现的位置
unordered_map<int, bool> appear; // 出现过的数字
unordered_map<int, int> dist; // 该数字的最小距离
unordered_set<int> cand;

int minimumSeconds(vector<int>& nums) {
    int n = nums.size();
    int ans = 1000000;
    for (int i = 0; i < n; i++) {
        int ele = nums[i];
        if (!appear[ele]) {
            cand.insert(ele);
            first[ele] = last[ele] = i;
            appear[ele] = true;
            continue;
        }
        dist[ele] = max(dist[ele], i - last[ele]);
        last[ele] = i;
    }
    for (auto ele : cand) {
        dist[ele] = max(dist[ele], first[ele] + n - last[ele]);
        ans = min(ans, dist[ele] / 2);
    }
}
```

```
return ans;
```

```
}
```