

解题报告
参考答案
<pre>#include<iostream> #define N 110 using namespace std; int n,d[N],cnt[10]; int main(){ cin>>n; for(int i=0;i<n;i++)cin>>d[i]; fill(cnt,cnt+10,0); int sum=0,ans=n+1; int tail=0,head=0; while(1){ while(head<n&&sum<7){ //头部移动 int ball=d[head++]; if(ball>7)continue; cnt[ball]++; if(cnt[ball]==1)sum++; } if(sum<7)break; ans=min(ans,head-tail); int ball=d[tail++]; if(ball>7)continue; cnt[ball]--; if(cnt[ball]==0)sum--; //尾部移动 } if(ans>n)cout<<"bye dragon"<<endl; else cout<<ans<<endl; return 0; }</pre>
题目分析
题目考察蠕动区间的理解和运用，需要注意的是：题目存在无解的情况。
提示
<p>题目是课件例题“收集三原色”的变化，可参考课件第 18 页的程序。因为龙珠编号是数字，不像“收集三原色”中颜色是字符串，所以不需要用 map，直接使用整数数组即可。</p> <p>本问题是最小优化问题，答案 ans 初始为 n+1。每种龙珠（1~9）的数量用计数器数组存放，龙珠种类数用另一个计数器存放。当遇到 8~9 星龙珠时忽略。当蠕动区间的头部向右移动遇到 1~7 星龙珠，且区间内该种龙珠数量由 0 变为 1 时（说明是唯一一颗），则龙珠种类计数器加 1。当蠕动区间的尾部向右移动经过 1~7 星龙珠，若该龙珠离开区间后数量变为 0（说明这颗龙珠是之前区间内同种类的唯一一颗），则龙珠种类计数器减 1。蠕动过程中若达成问题要求（凑足了 7 种龙珠），则记录最优解。若所有区间都不能达成 7 种龙珠（ans 仍为 n+1），则输出 bye dragon。</p>
易错点
1) 注意蠕动区间头部越界的情形。

- 2) 搞清楚++是后自增还是前自增。
- 3) 题目存在无解的情况，输出"bye dragon"。

解题报告
参考答案
<pre> #include<iostream> #define N 1010 #define R 205 using namespace std; int n,k,d[N],cnt[R]; int main(){ cin>>n>>k; for(int i=0;i<n;i++)cin>>d[i]; int sum=0,ans=n+1; int tail=0,head=0; while(1){ while(head<n&&sum<k){ //头部移动 int country=d[head++]; cnt[country]++; if(cnt[country]==1)sum++; } if(sum<k)break; //终止条件 ans=min(ans,head-tail); int country=d[tail++]; cnt[country]--; if(cnt[country]==0)sum--; //尾部移动 } if(ans>n)cout<<"impossible"<<endl; else cout<<ans<<endl; return 0; } </pre>
题目分析
<p>题目考察蠕动区间的理解和运用，需要注意的是：题目存在无解的情况。</p>
提示
<p>题目是课件例题“收集三原色”的变化，可参考课件第 18 页的程序。</p> <p>本问题是最小优化问题，答案 ans 初始为 n+1。每个国家经过的次数用计数器数组存放，经过国家种类数用另一个计数器存放。当蠕动区间的头部向右移动遇到某个国家，该国家经过次数加 1。特别地，若加 1 是由 0 变为 1，则国家种类计数器加 1。当蠕动区间的尾部向右移动时，某个国家离开区间后若经过次数变为 0，则国家种类计数器减 1。蠕动过程中若达成问题要求（经过了 k 个不同国家），则记录最优解。若所有区间都不能达成 k 个国家（ans 仍为 n+1），则输出 impossible。</p>
易错点
<ol style="list-style-type: none"> 1) 注意蠕动区间头部越界的情形。 2) 搞清楚++是后自增还是前自增。 3) 题目存在无解的情况，输出"impossible"。

解题报告
参考答案
<pre>#include<iostream> #define N 1010 #define R 205 using namespace std; int n,k,d[N],cnt[R]; int main(){ cin>>n; for(int i=0;i<n;i++){ cin>>d[i]; cnt[d[i]]++; k+=(cnt[d[i]]==1); } fill(cnt,cnt+R,0); int sum=0,ans=n+1; int tail=0,head=0; while(1){ while(head<n&&sum<k){ //头部移动 int country=d[head++]; cnt[country]++; if(cnt[country]==1)sum++; } if(sum<k)break; //终止条件 ans=min(ans,head-tail); int country=d[tail++]; cnt[country]--; if(cnt[country]==0)sum--; //尾部移动 } if(ans>n)cout<<"impossible"<<endl; else cout<<ans<<endl; return 0; }</pre>
题目分析
题目考察蠕动区间的理解和运用，需要注意的是：问题获得满足的国家数需要计算出来。
提示
<p>题目是“491 免费旅游 1”的变化。</p> <p>问题获得满足的国家数 k，是在读取游轮每周经过国家时计算出来的，就是不同国家的数量。</p> <p>本问题是最小优化问题，答案 ans 初始为 $n+1$。每个国家经过的次数用计数器数组存放，经过国家种类数用另一个计数器存放。当蠕动区间的头部向右移动遇到某个国家，该国家经过次数加 1。特别地，若加 1 是由 0 变为 1，则国家种类计数器加 1。当蠕动区间的尾部向右移动时，某个国家离开区间后若经过次数变为 0，则国家种类计数器减 1。蠕动过程中若达成问题要求（经过了 k 个不同国家），则记录最优解。若所有区间都不能达成 k</p>

个国家 (ans 仍为 $n+1$), 则输出 impossible。
易错点
<ul style="list-style-type: none">1) 注意蠕动区间头部越界的情形。2) 搞清楚++是后自增还是前自增。3) 题目存在无解的情况, 输出"impossible"。