

信奥算法

复习

股神四

股神四

你是股神，能看到未来 n 天股票盈利或亏损，第 i 天盈利 $x[i]$ 元，若 $x[i]$ 是负数，代表亏损。可挑选一段不超过 k 天连续投资，请计算最多盈利。不投资时盈利为0.

输入 n 和 k ，第二行为整数 $x[1], x[2], \dots, x[n]$ 。 $1 \leq k \leq n \leq 200000$, $|x_i| \leq 10000$ 。输出一个整数

输入样例

4 3
2 -1 2 2

输出样例

4

输入样例

4 2
-3 9 5 1

输出样例

14

暴力该怎么做?

连续子段最大和： 长度不超过k

$f[i]$ 代表以*i*号结尾的**长度不超过k**的连续和最大值

计算 $f[i]$ 时考虑k种可能

$$f[i] = \max \left\{ \begin{array}{l} x[i], \\ x[i] + x[i-1], \\ x[i] + x[i-1] + x[i-2], \\ \dots \dots \dots \\ x[i] + x[i-1] + \dots + x[i-k+1] \end{array} \right\}$$

长度为1

长度为2

长度为3

长度为k

注意越界

连续子段最大和： 长度不超过k

$f[i]$ 代表以*i*号结尾的**长度不超过k**的连续和最大值

计算 $f[i]$ 时考虑k种可能

$$f[i] = \max \left\{ \begin{array}{l} s[i] - s[i-1], \\ s[i] - s[i-2], \\ s[i] - s[i-3], \\ \dots \dots \dots \\ s[i] - s[i-k] \end{array} \right\}$$

连续和
转换为
前缀和

长度为1

长度为2

长度为3

长度为k

注意越界

连续子段最大和： 长度不超过k

$f[i]$ 代表以*i*号结尾的**长度不超过k**的连续和最大值

计算 $f[i]$ 时考虑k种可能

$$f[i] = s[i] - \min \left\{ \begin{array}{l} s[i-1], \\ s[i-2], \\ s[i-3], \\ \dots \dots \dots \\ s[i-k] \end{array} \right\}$$

固定*i*时
前缀和数组
窗口最小值

单调队列

```
6 cin>>n>>k;
7 for(int i=1;i<=n;i++)cin>>x[i];
8 for(int i=1;i<=n;i++)s[i]=s[i-1]+x[i];
9 int l=0,r=0;
10 for(int i=0;i<=n;i++){
11     while(l<r&& i-q[l]>=k)l++;
12     while(l<r&& s[i]<s[q[r-1]])r--;
13     q[r++]=i;
14     MIN[i]=s[q[l]];
15 }
16 for(int i=1;i<=n;i++)f[i]=s[i]-MIN[i-1];
17 cout<<max(0,*max_element(f+1,f+n+1))<<endl;
```

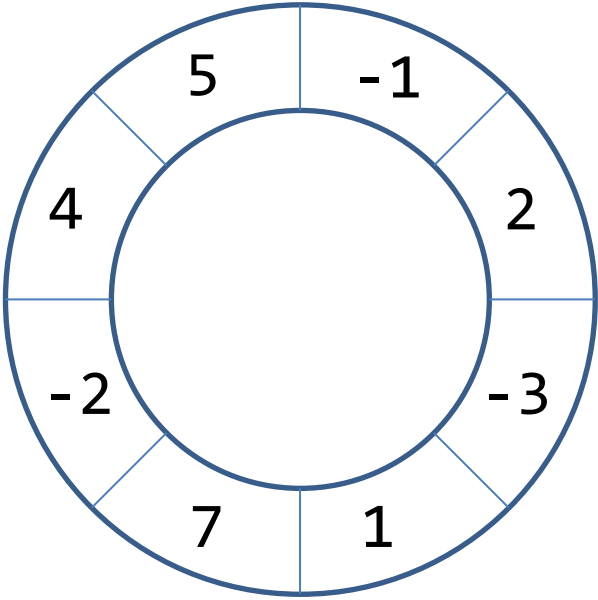
快快编程736

快快编程
kkcoding.net

方法1

断环+拉直+克隆

断环+拉直+克隆



-1	2	-3	1	7	-2	4	5	-1	2	-3	1	7	-2	4	5
----	---	----	---	---	----	---	---	----	---	----	---	---	----	---	---

```
8  cin>>n>>m;
9  for(int i=1;i<=n;i++)cin>>x[i];
10 for(int i=n+1;i<=n*2;i++)x[i]=x[i-n];
11 s[0]=0;
12 for(int i=1;i<=n*2;i++)s[i]=s[i-1]+x[i];

13 int ans=s[m];
14 for( )
15     ans=max(ans,s[i]-s[i-m]);
16 cout<<ans<<endl;
```

断环+拉直+克隆

正整数 n, m ,
 $1 \leq m \leq n \leq 100000$

```
3 const int N=;  
4 int n,m,s[N],x[N];
```

高频易错点

环形问题克隆时
数组大小需要翻倍

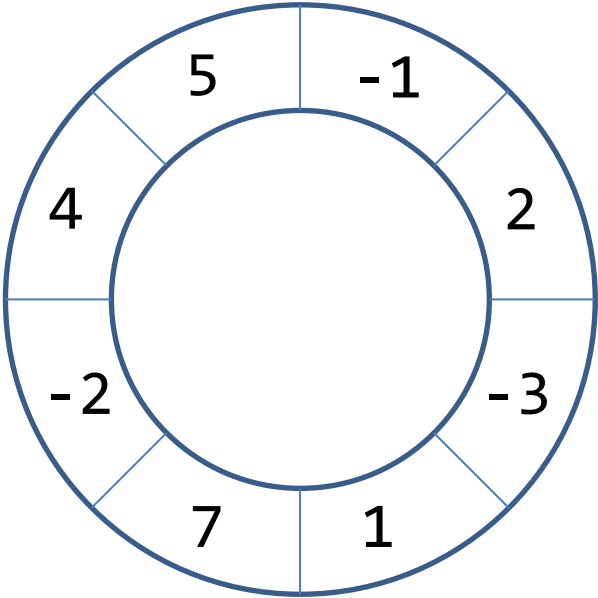
方法2

首尾情况分类

首尾不相连

首尾相连

首尾情况分类



首尾不相连

-1	2	-3	1	7	-2	4	5
----	---	----	---	---	----	---	---

首尾相连

-1	2	-3	1	7	-2	4	5
----	---	----	---	---	----	---	---

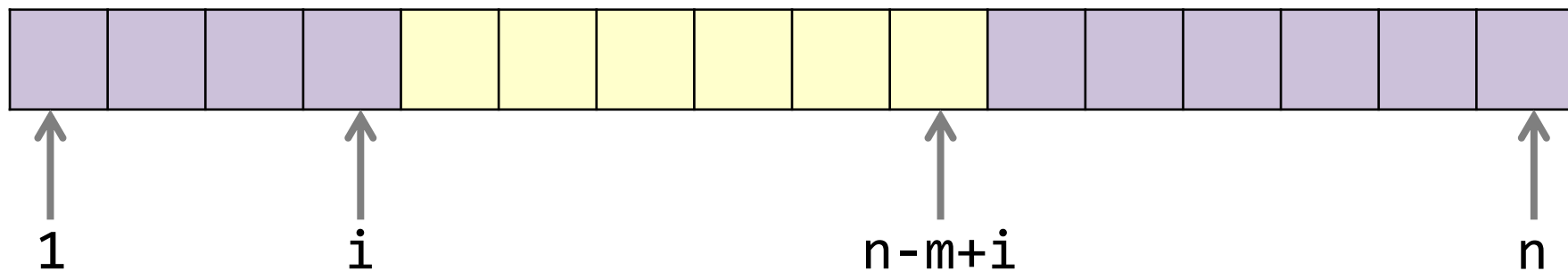
```
8  cin>>n>>m;
9  for(int i=1;i<=n;i++)cin>>x[i];
10 for(int i=1;i<=n;i++)s[i]=s[i-1]+x[i];

11 int ans=s[m];
12 for(int i=m+1;i<=n;i++)
13     ans=max(ans,s[i]-s[i-m]);

14 for(int i=1;i<=m-1;i++)
15     ans=max(ans, );
16 cout<<ans<<endl;
```

左侧共*i*个

右侧共*m-i*个



2种
解释

加法

$$s[i] + (s[n] - s[n-m+i])$$

减法

$$s[n] - (s[n-m+i] - s[i])$$

补集转换

快快编程2472

快快编程
kkcoding.net

环形问题2种 典型思路是什么？

方法1

断环+拉直+克隆

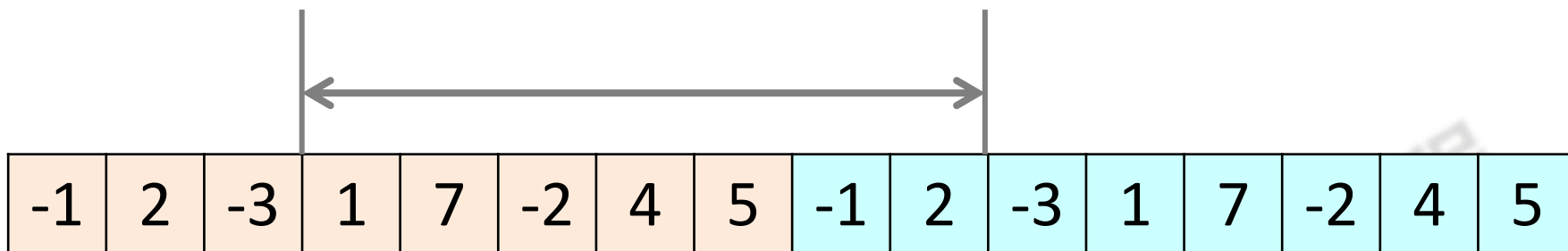
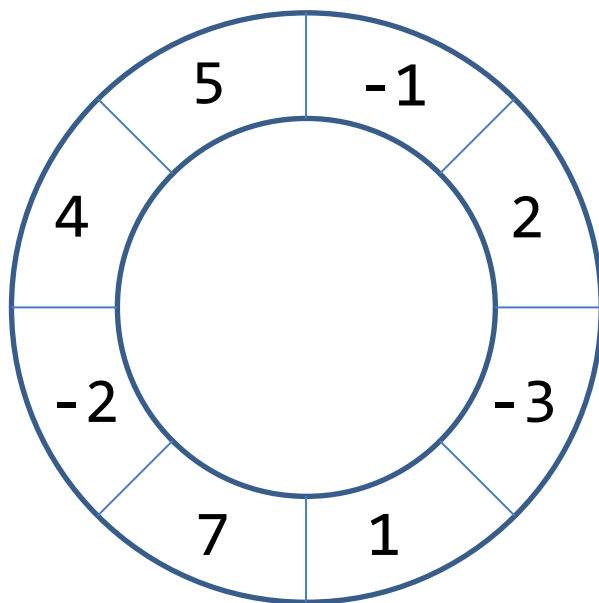
方法2

首尾情况分类

方法1

断环+拉直+克隆

断环+拉直+克隆



最大子段和问题的变种

子段长度不能超过n

单调
队列

断环+拉直+克隆

```
8  cin>>n;  
9  for(int i=1;i<=n;i++)cin>>x[i];  
10 for(int i=n+1;i<=n*2;i++)x[i]=x[i-n];  
11 for(int i=1;i<=n*2;i++)s[i]=s[i-1]+x[i];
```

断环+拉直+克隆

```
12 int k=n;  
13 int l=0,r=0;  
14 for( ) {  
15     while(l<r&& i-q[l]>=k) l++;  
16     while(l<r&& s[i]<s[q[r-1]]) r--;  
17     q[r++]=i;  
18     MIN[i]=  
19 }
```

断环+拉直+克隆

```
20 for(int i=1;i<=n*2;i++)f[i]=s[i]-MIN[i-1];  
21 int ans=*max_element(  
22 cout<<ans<<endl;
```

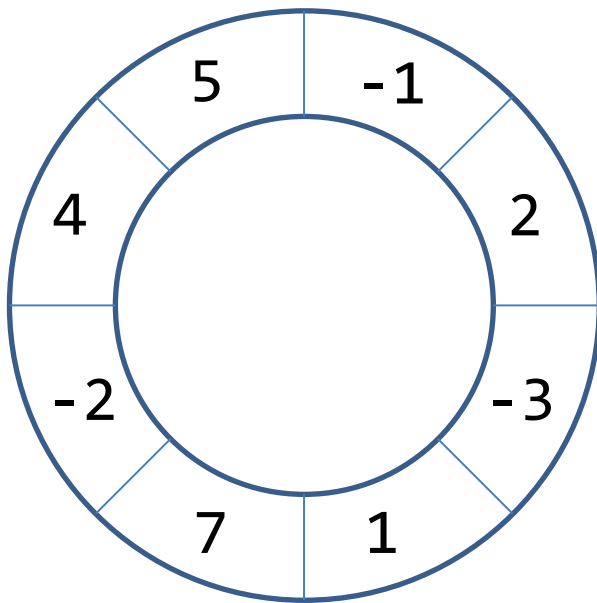
方法2

首尾情况分类

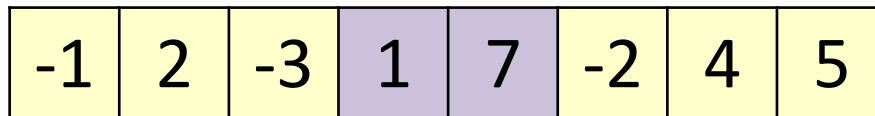
首尾不相连

首尾相连

首尾情况分类

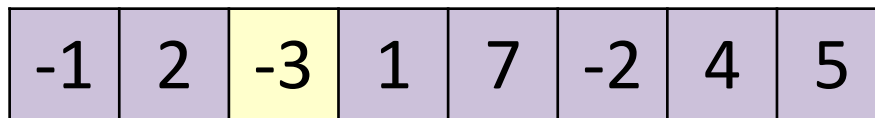


首尾不相连



最大子段和

首尾相连



最小子段和

首尾情况分类

```
14  int sum=0;
15  for(int i=1;i<=n;i++)sum+=x[i];

16  f[1]=x[1];
17  for(int i=2;i<=n;i++)f[i]=max(f[i-1],0)+x[i];
18  g[1]=x[1];
19  for(int i=2;i<=n;i++)g[i]=min(g[i-1],0)+x[i];

20  int ans1=*max_element(f+1,f+n+1);
21  int ans2=
22  int ans=max(ans1,ans2);
23  cout<<ans<<endl;
```

快快编程2473

快快编程
kkcoding.net

环形问题2种 典型思路是什么？

方法1

断环+拉直+克隆

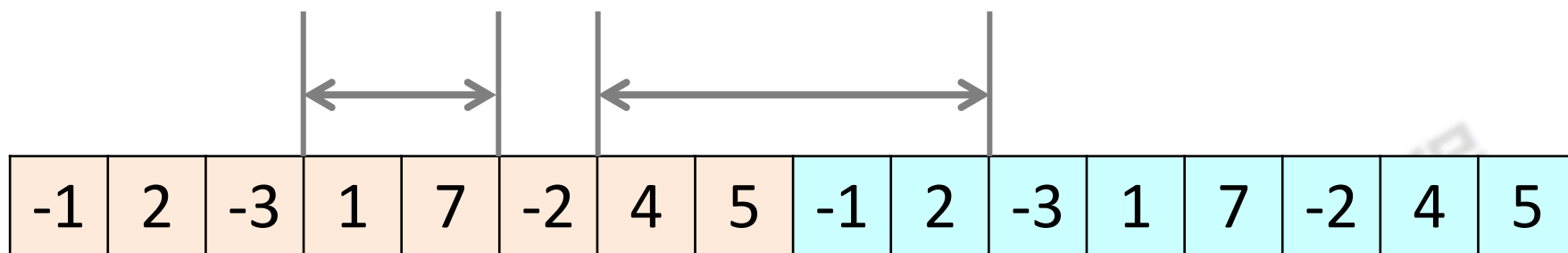
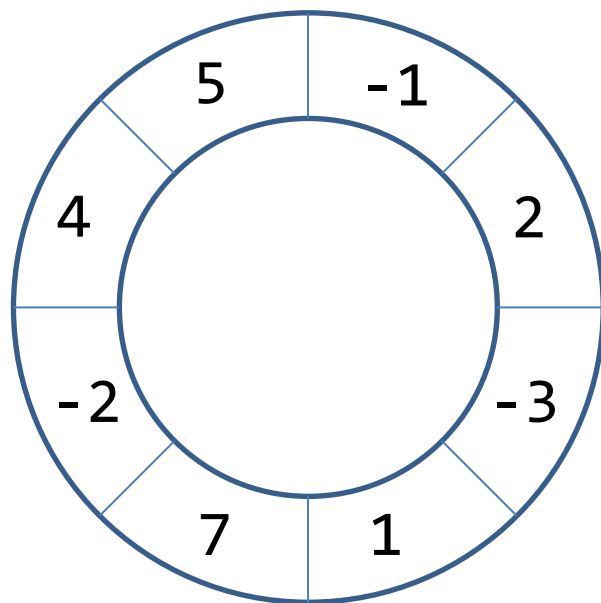
方法2

首尾情况分类

方法1

断环+拉直+克隆

断环+拉直+克隆



最大2子段和问题的变种

跨度不能超过n

暴力
解法

断环+拉直+克隆

```
9  cin>>n;
10 for(int i=1;i<=n;i++)cin>>x[i];
11 for(int i=n+1;i<=n*2;i++)x[i]=x[i-n];
12 for(int i=1;i<=n*2;i++)s[i]=s[i-1]+x[i];
13 int ans=-INF;
14 for(int a=1;a<=n;a++)
15     for(int b=a;b<=a+n-1;b++)
16         for(int c=)
17             for(int d=c;d<=a+n-1;d++)
18                 ans=max(ans,);
19 cout<<ans<<endl;
```

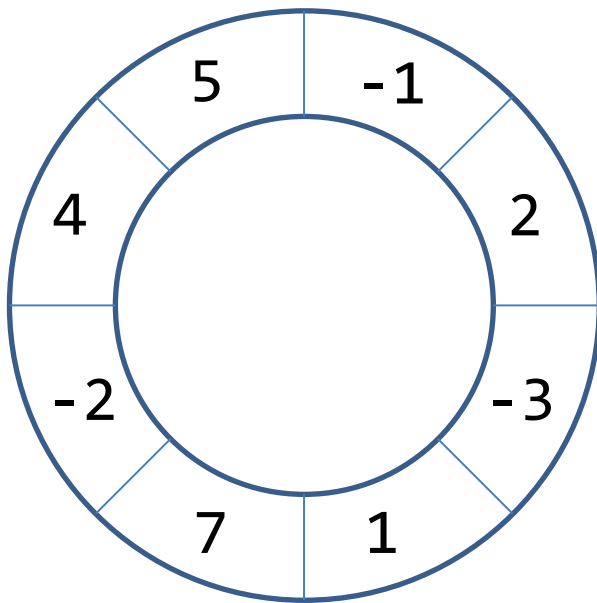
方法2

首尾情况分类

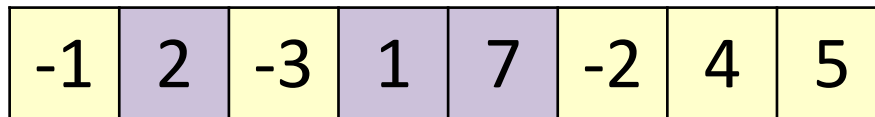
首尾不相连

首尾相连

首尾情况分类

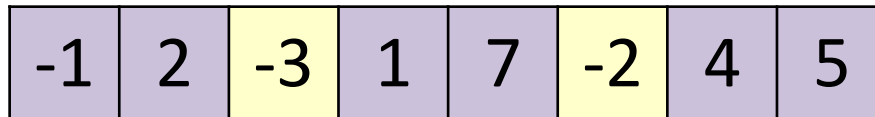


首尾不相连



最大2段和

首尾相连



最小2段和

```
37     cin>>n;
38     for(int i=1;i<=n;i++)cin>>x[i];
39     int sum=0;
40     for(int i=1;i<=n;i++)sum+=x[i];
41     solveMax();
42     solveMin();
43     int ans=-INF;
44     for(int i=1;i<=n-1;i++)
45         ans=max(ans,preMax[i]+postMax[i+1]);
46     for(int i=1;i<=n-1;i++)
47         ans=max(ans,sum-preMin[i]-postMin[i+1]);
48     cout<<ans<<endl;
```

preMin[i]表示以前i号里最小子段和

postMin[i]表示以i号到n号里最小子段和

leftMax[i]表示以i号结尾的最大子段和

rightMax[i]表示以i号开始的最大子段和

```
14 void solveMax(){
15     leftMax[1]=x[1];
16     for(int i=2;i<=n;i++)leftMax[i]=max(leftMax[i-1],0)+x[i];
17     preMax[1]=leftMax[1];
18     for(int i=2;i<=n;i++)preMax[i]=max(preMax[i-1],leftMax[i]);
19     rightMax[n]=x[n];
20     for(int i=n-1;i>=1;i--)rightMax[i]=max(rightMax[i+1],0)+x[i];
21     postMax[n]=rightMax[n];
22     for(int i=n-1;i>=1;i--)postMax[i]=max(postMax[i+1],rightMax[i]);
23 }
```

preMax[i]表示以前i号里最大子段和

postMax[i]表示以i号到n号里最大子段和

快快编程作业

736

2472

2473