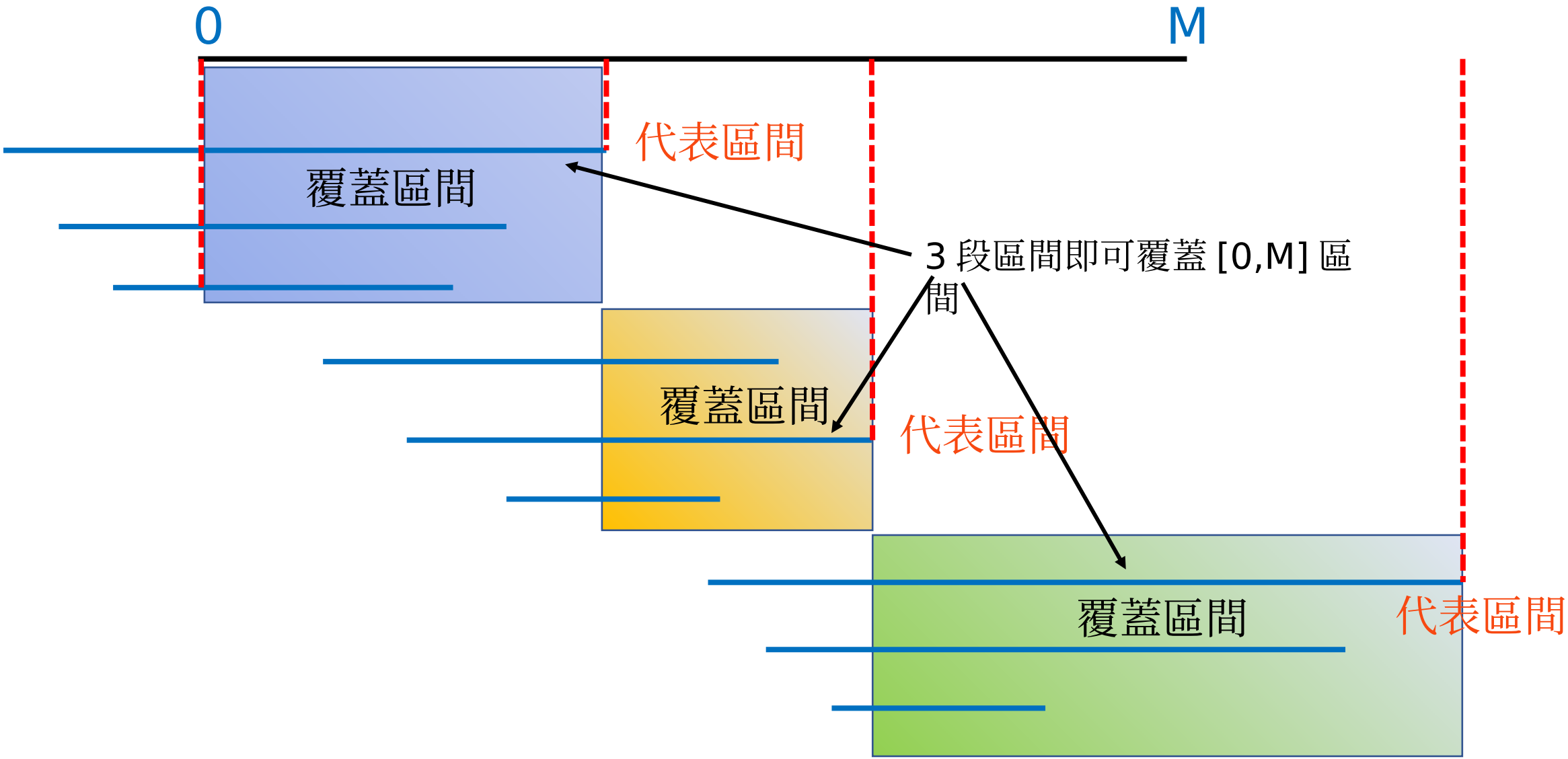
UVa 10020 Minimal coverage

UVa 10020 Minimal coverage (Time Limit: 3 seconds)

最少個的覆蓋線段：給定正整數 M (1 ≤ M ≤ 5000) 與多組線段 (區間) 座標 [Li, Ri] ， [Li, Ri](在 X 軸上) 。請選擇最少數量的線段，使得選擇的線段完全覆蓋 [0, M] 。
請輸出最少的線段數與覆蓋 [0,M] 的各個線段 ; 如果無解 , 請輸出 0 。


0
M
代表區間
覆蓋區間
3 段區間即可覆蓋 [0,M] 區間
覆蓋區間
代表區間
覆蓋區間
代表區間

Sample Input

2
空白行
1
-1 0
-5 -3
2 5
0 0
空白行
1
-1 0
0 1
0 0

Test Case 數目

M 值

線段 (區間) 資訊

Test Case 輸入結束

Test Case #1

Test Case #2

Sample Output

0
空白行

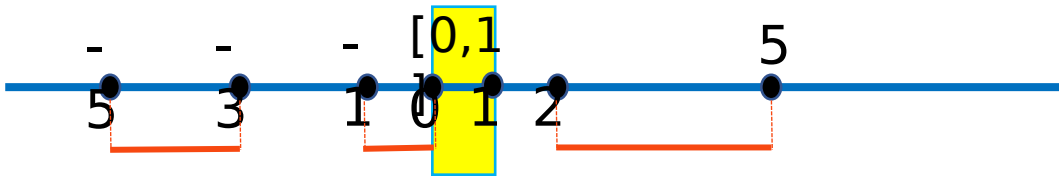
(無解)

1
0 1

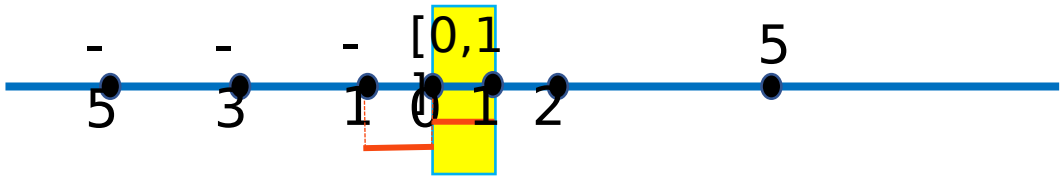
覆蓋最少的線段數

覆蓋的線段

Test Case #1 M= 1



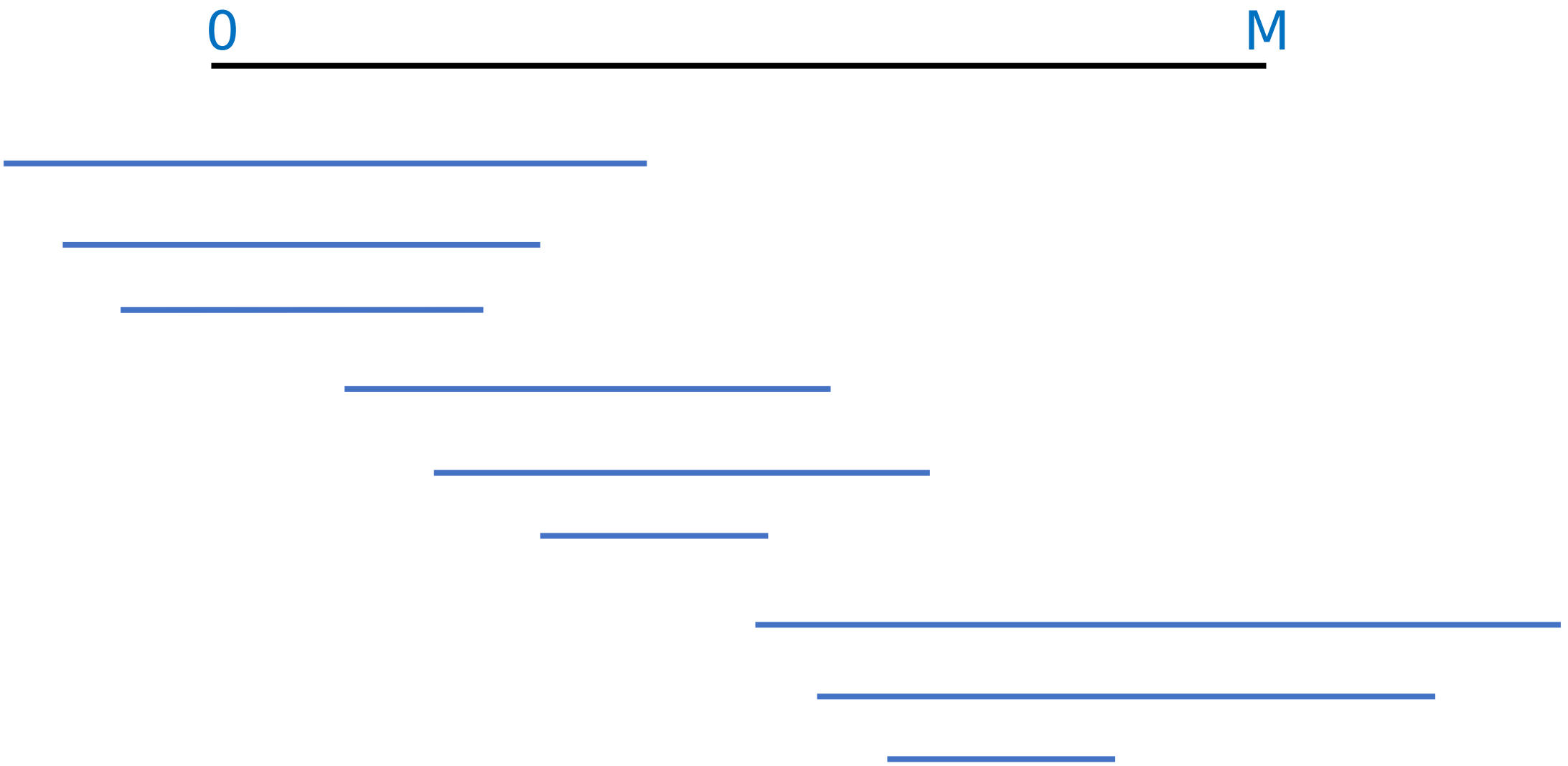
Test Case #2 M= 1

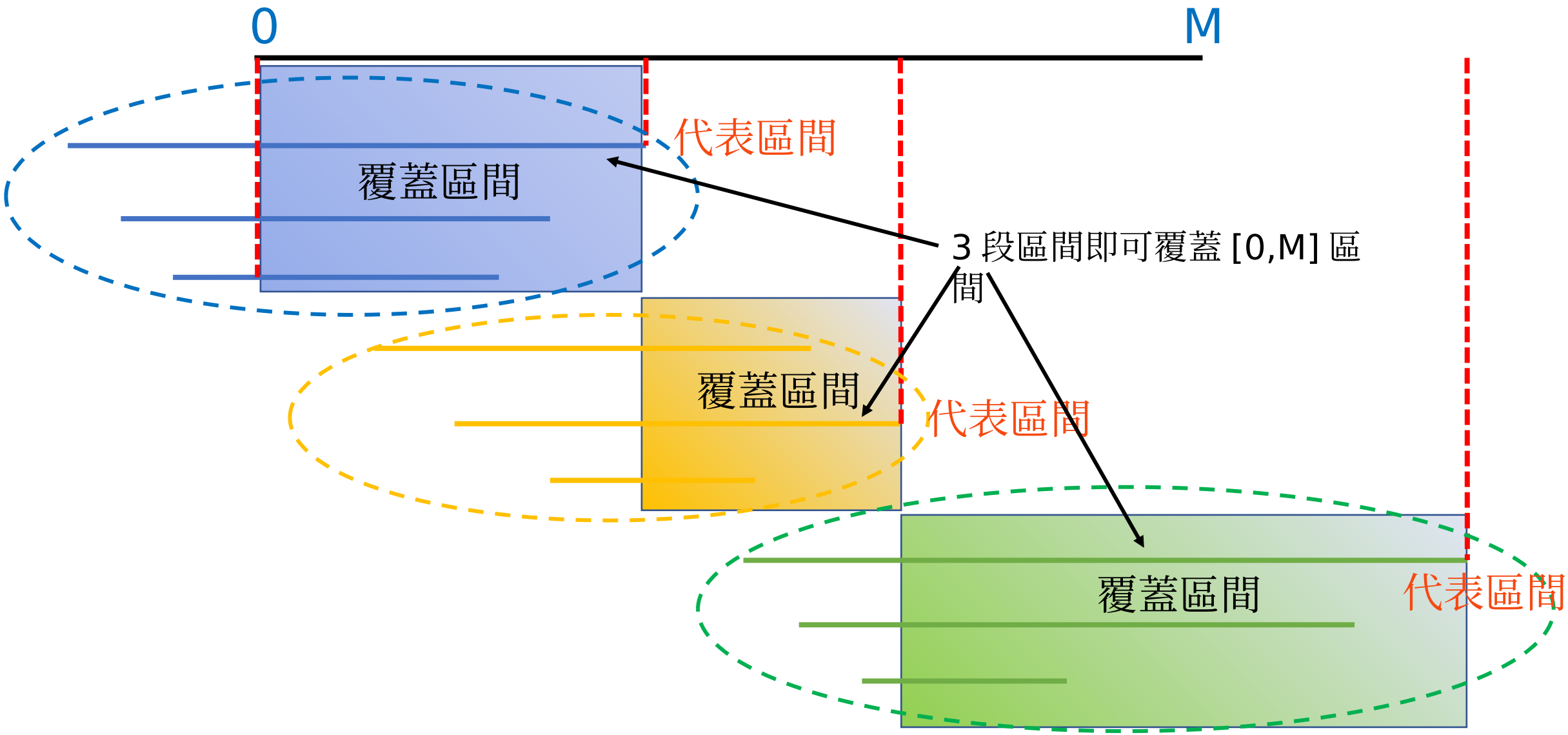


Solution

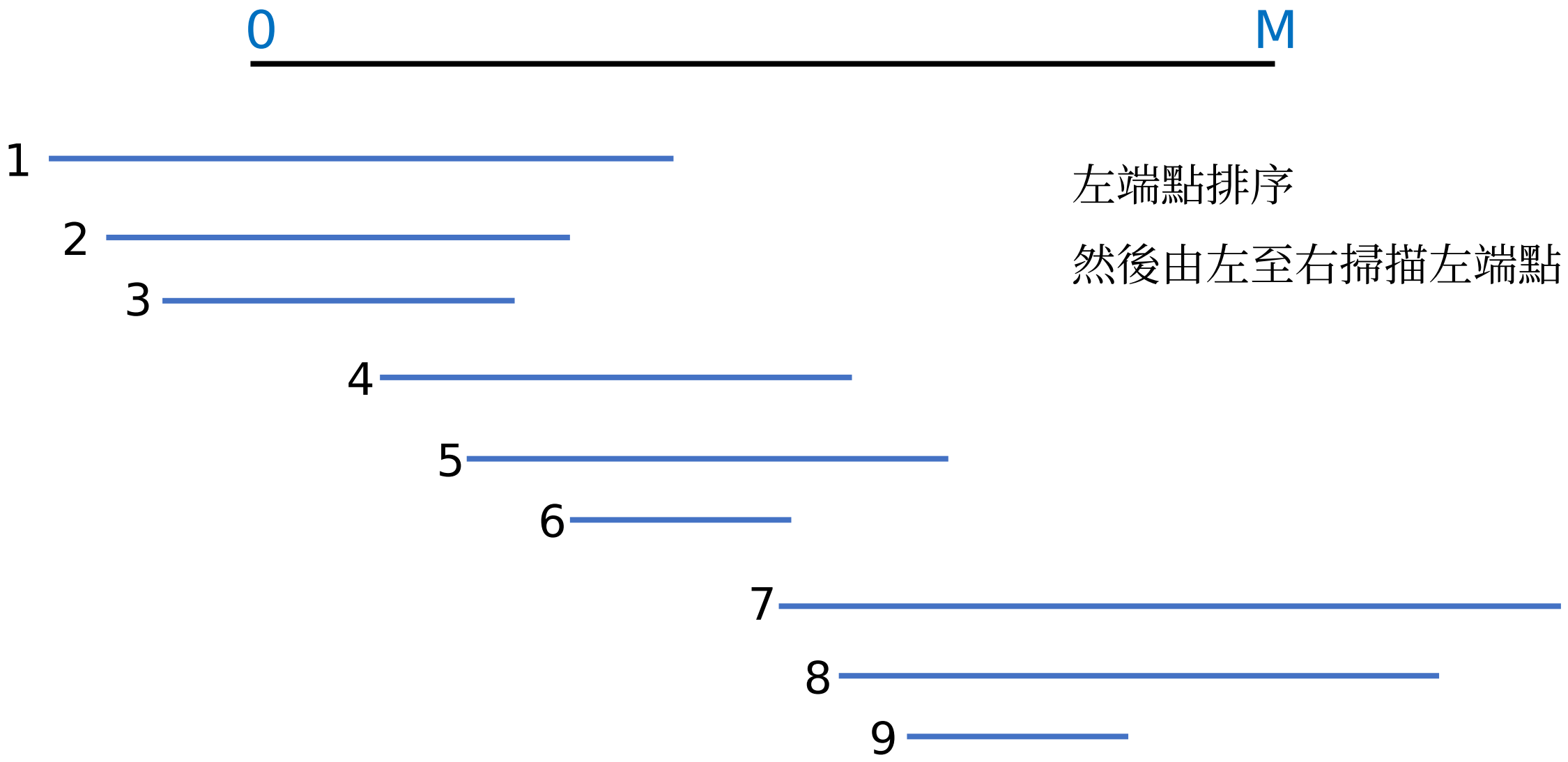
- Interval Covering 問題
- Greedy Method
- 由左至右掃描每個區間左端點 , 依序找出一團一團重疊在一起的區間 , 每一團的覆蓋範圍為 [left,right], 其 right 值是重疊區間中右端點值最大者 ; 下一團的 left 值就是前一團的 right 值 (第一團的 left 值是 0), 當 left M, 就停止掃描 .
- 當掃到下一左端點 , 它已經在當時該團 left 值右邊時 , 表示該團已經掃描結束 , 它的覆蓋範圍為當時的 [left,right], 下一團的 left 值就是前一團的 right 值 .
  - 如果 left=right 而且 right < M, 表示有一段範圍覆蓋不到 , 則無解
  - 如果最後一團之 right M , 表示有解 ; 不然無解 .

有解狀況


0
M

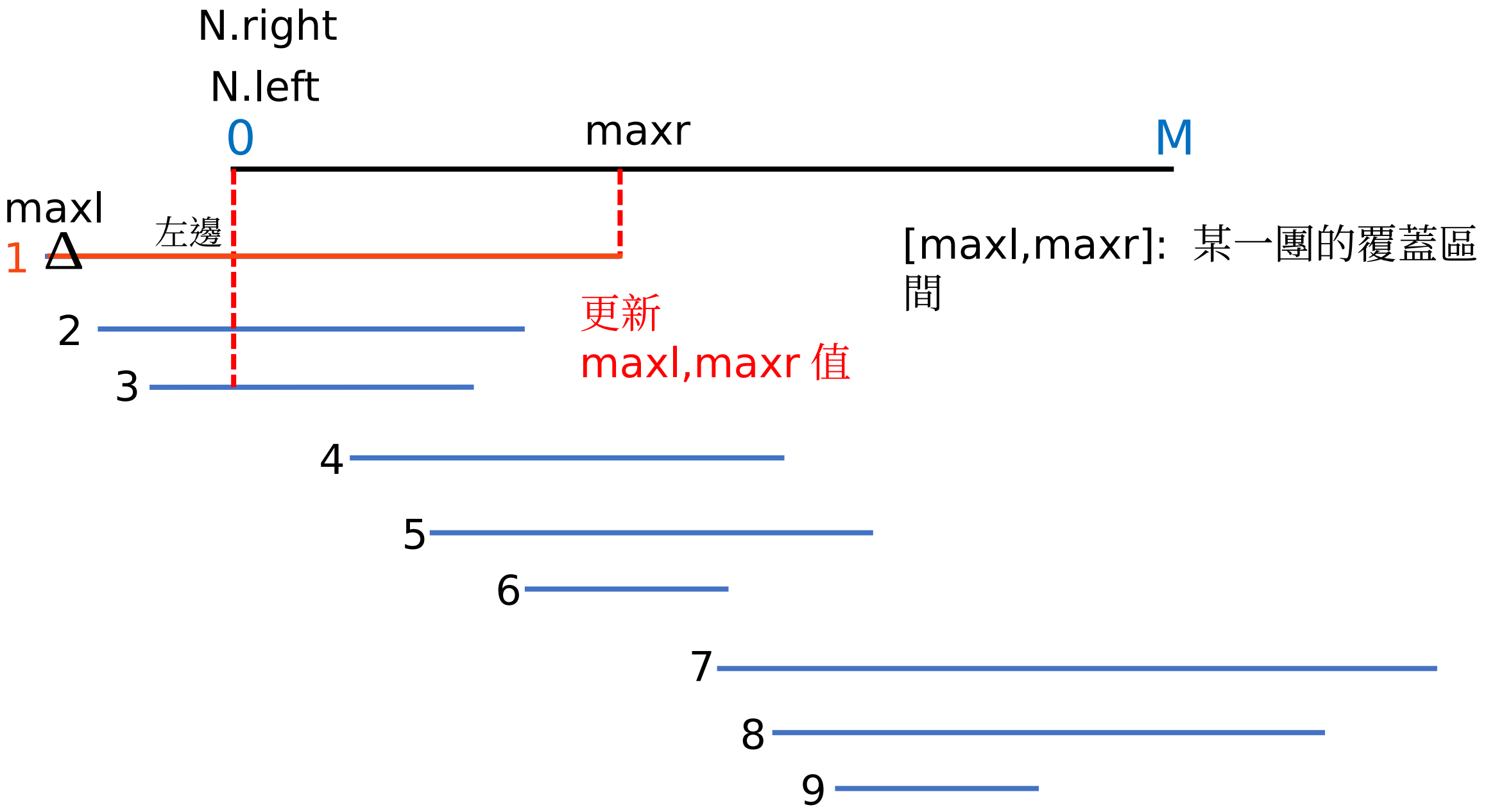

0
M
代表區間
覆蓋區間
3 段區間即可覆蓋 [0,M] 區間
覆蓋區間
代表區間
覆蓋區間
代表區間

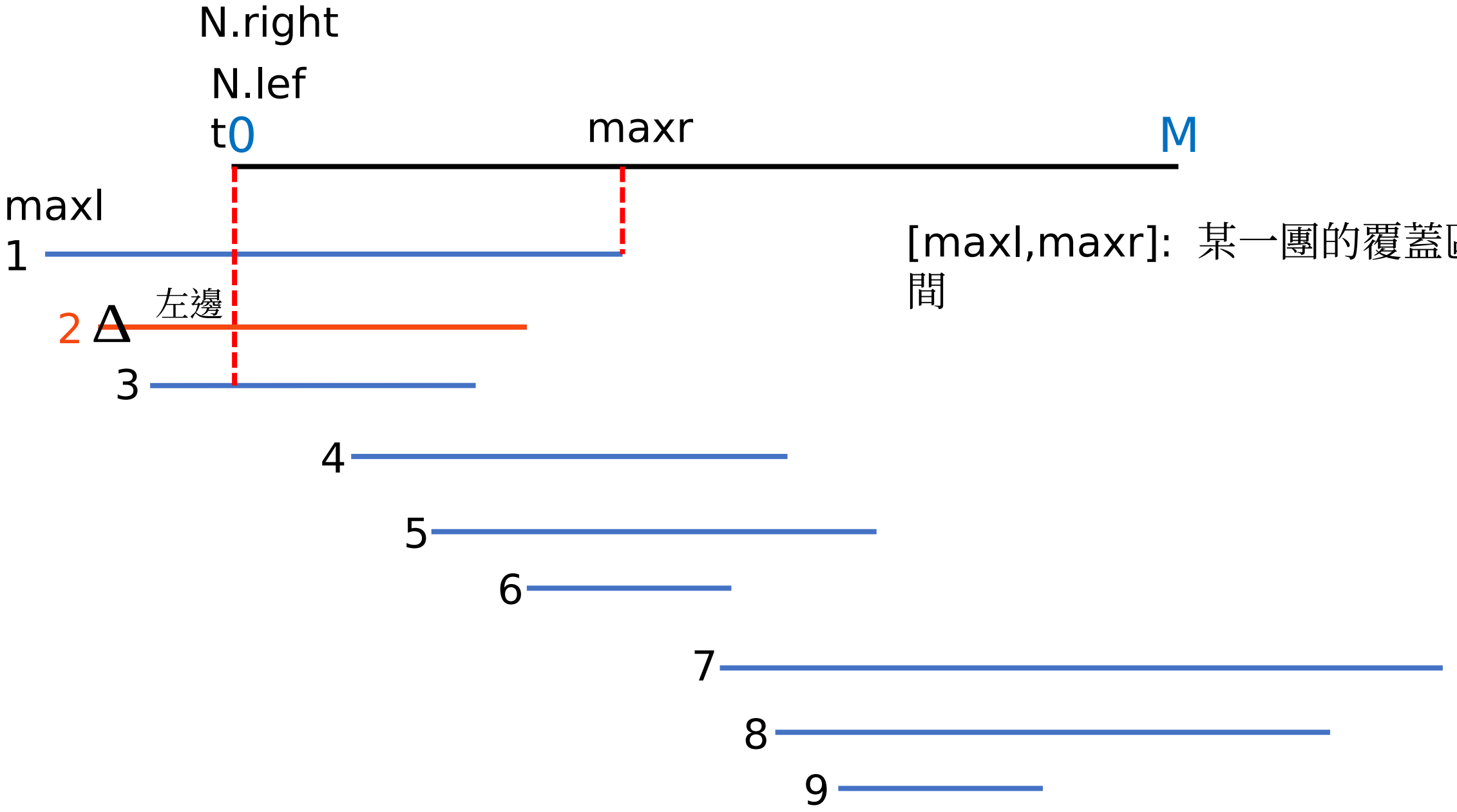
演算法

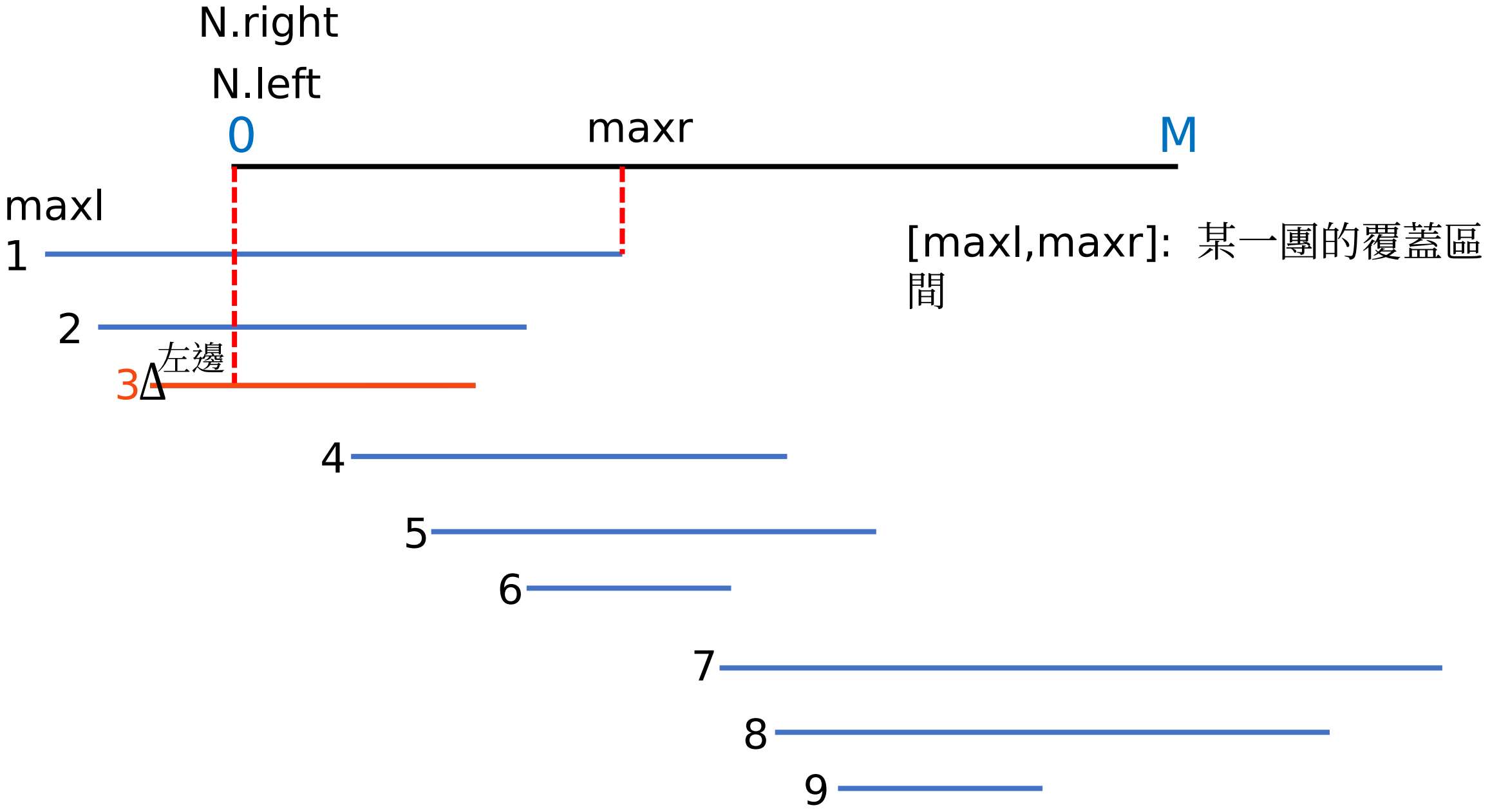


左端點排序

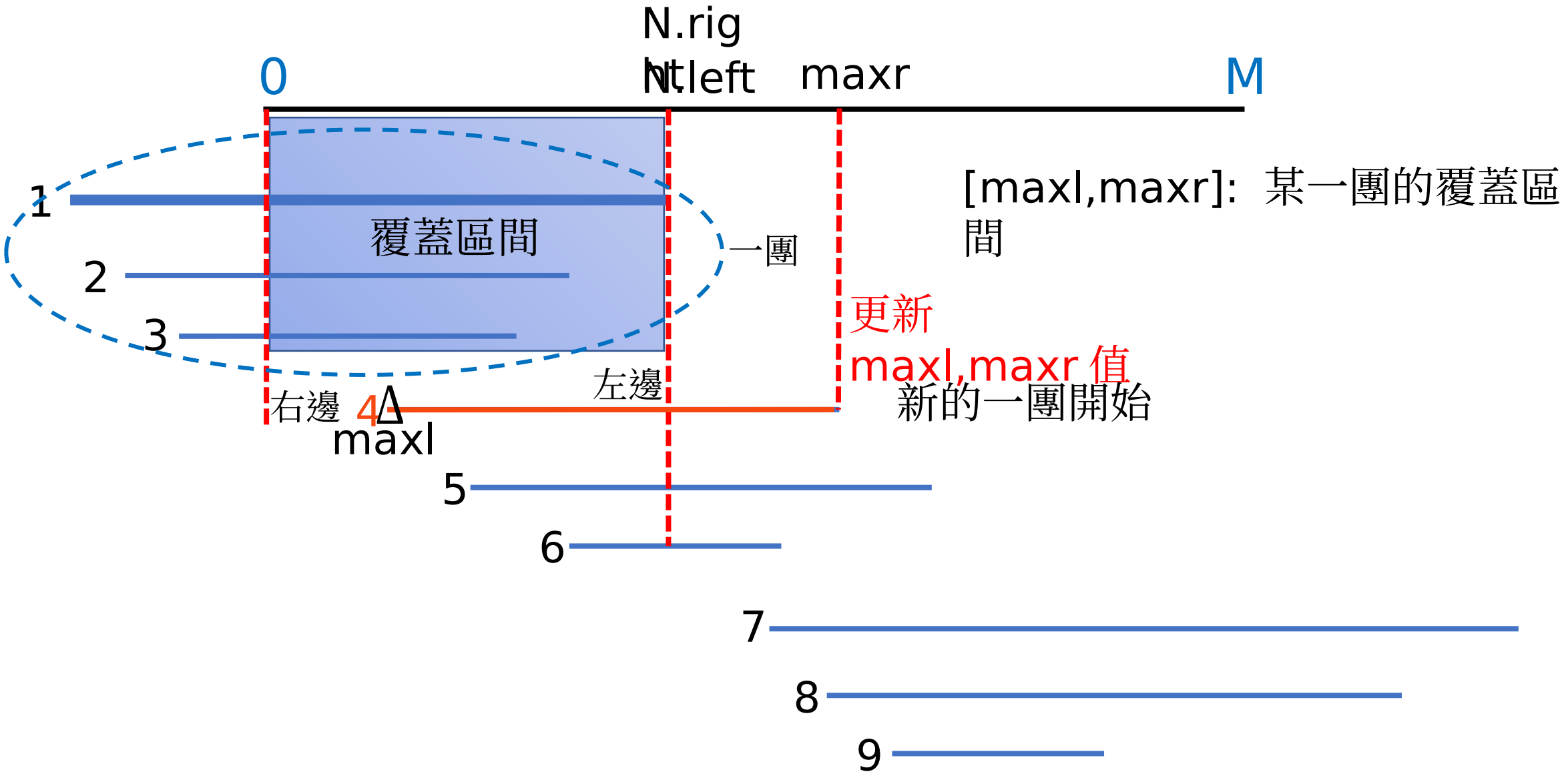
然後由左至右掃描左端點

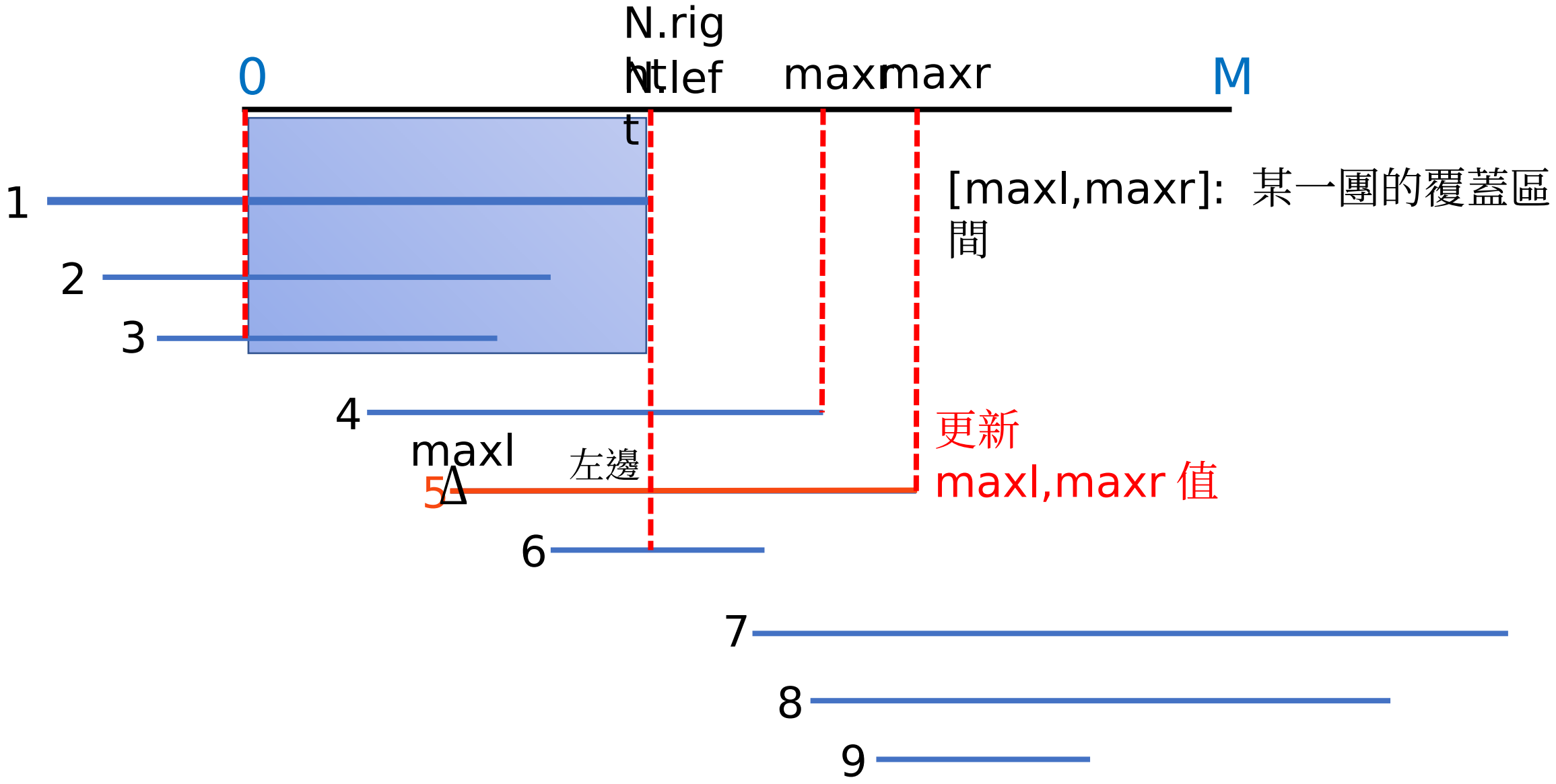

N.right
N.left
0
maxr
M
maxl
左邊
1
2
3
4
5
6
7
8
9
更新
maxl,maxr 值
[maxl,maxr]: 某一團的覆蓋區間

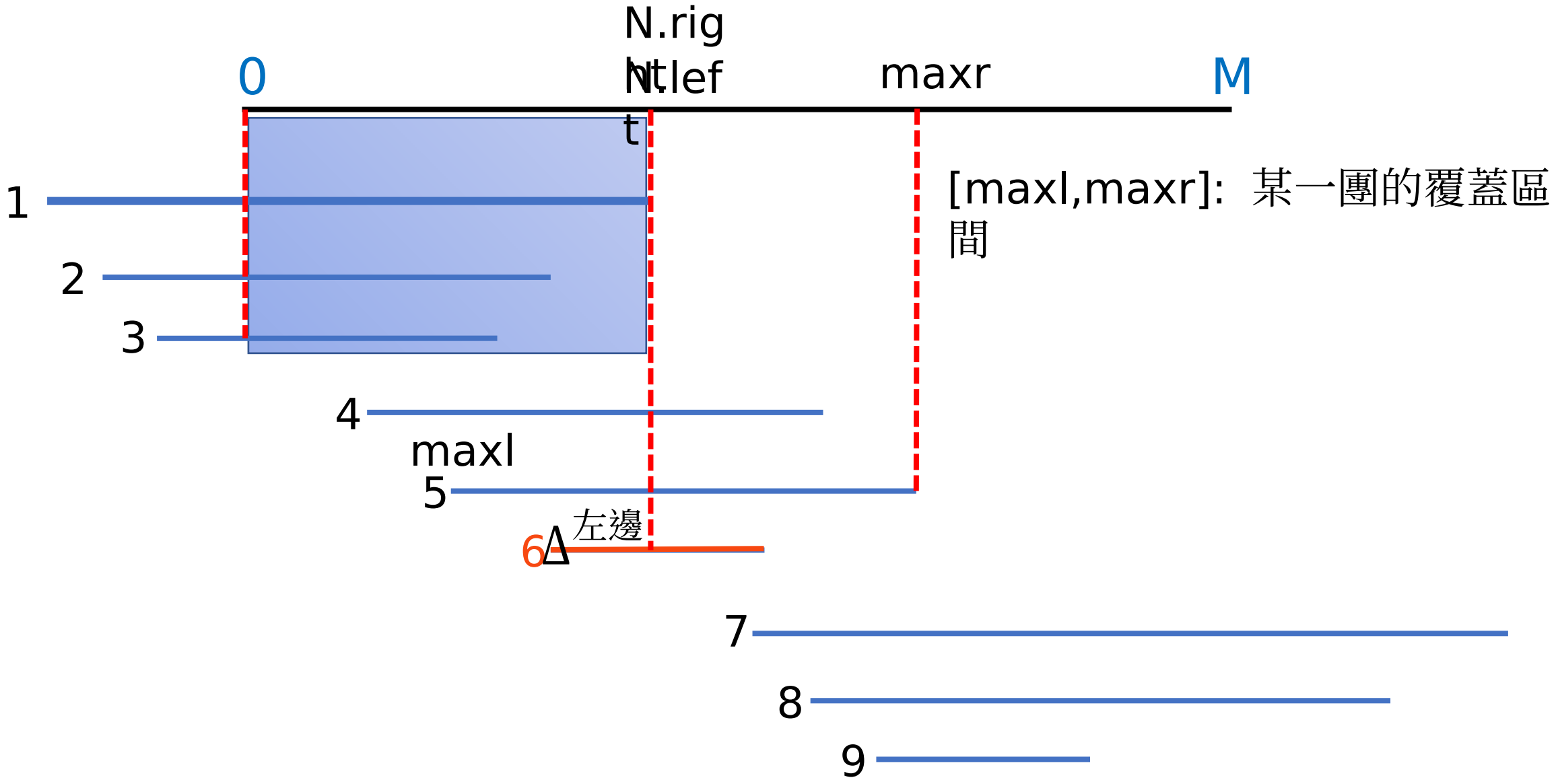

N.right
N.lef
t0
maxr
M
maxl
1
左邊
2 Δ
3
4
5
6
7
8
9
[maxl,maxr]: 某一團的覆蓋區間

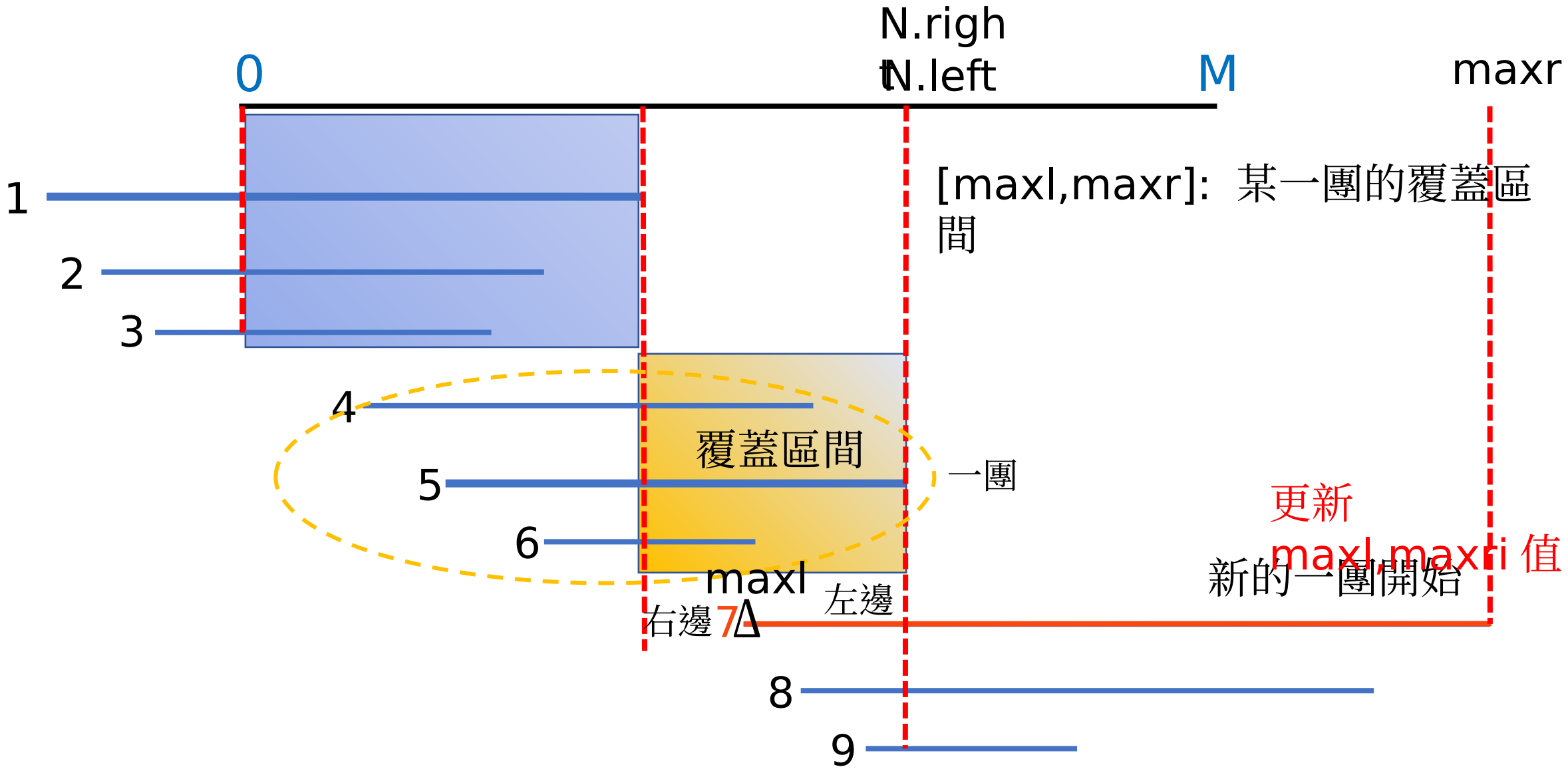


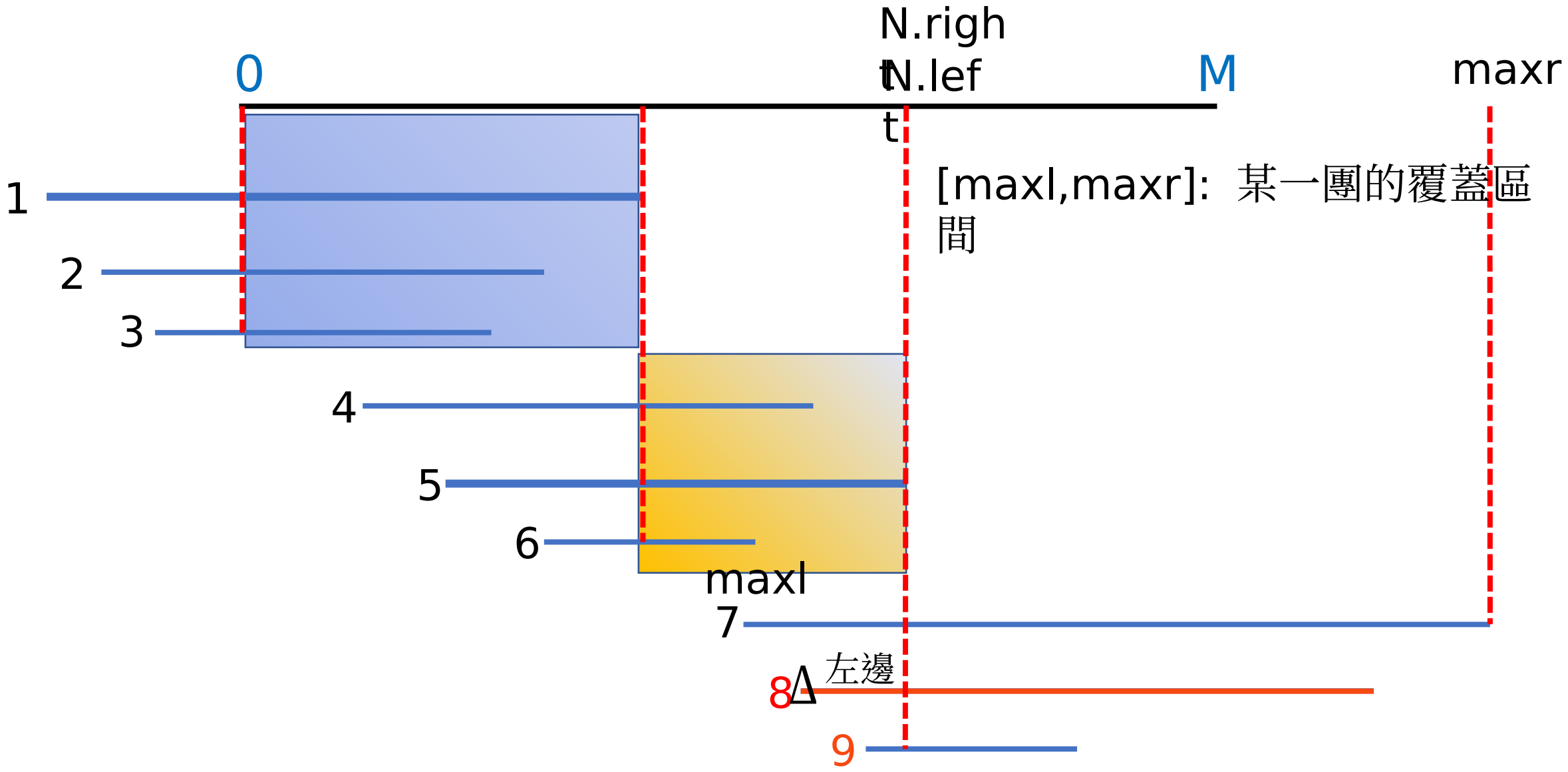
[maxl,maxr]: 某一團的覆蓋區間

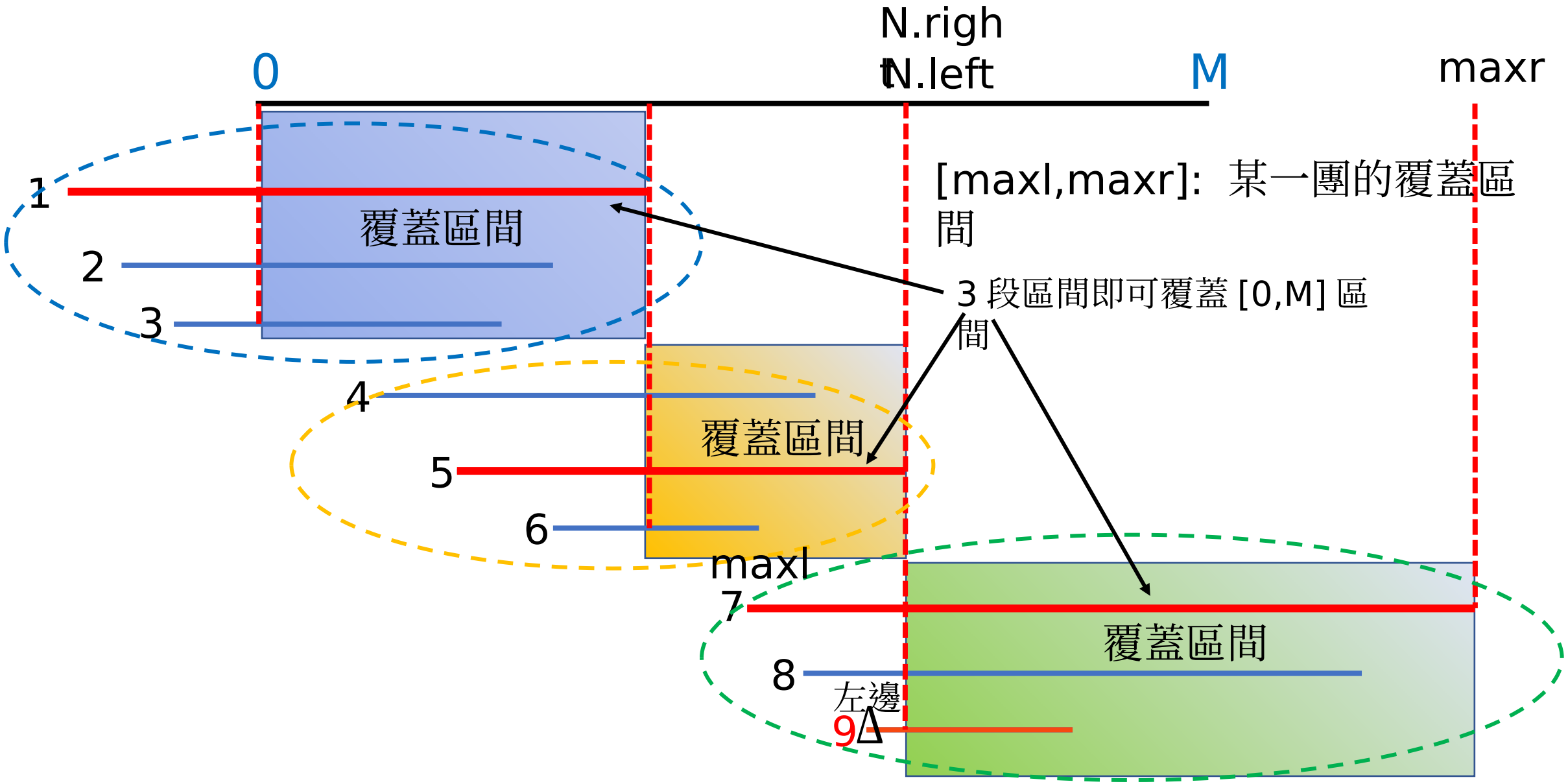

N.right
0
N.left
maxr
M
1
2
3
覆蓋區間
一團
[maxl,maxr]: 某一團的覆蓋區間
更新
maxl,maxr 值
新的一團開始
左邊
右邊
4
maxl
5
6
7
8
9


N.right
0
N.left
maxr
M
t
maxl
1
2
3
4
5
6
7
8
9
[maxl,maxr]: 某一團的覆蓋區間
更新
maxl,maxr 值
左邊


N.right
N.left
0
maxr
M
[maxl,maxr]: 某一團的覆蓋區間
1
2
3
4
maxl
5
左邊
6
7
8
9


N.righ
t
N.left
0
M
maxr
[maxl,maxr]: 某一團的覆蓋區間
1
2
3
4
5
6
覆蓋區間
一團
maxl
右邊
左邊
更新
maxl,maxr 值
新的一團開始
8
9


N.righ
t
0
N.lef
t
M
maxr
[maxl,maxr]: 某一團的覆蓋區間
1
2
3
4
5
6
maxl
7
左邊
8
Δ
9


0
N.right
N.left
M
maxr
[maxl,maxr]: 某一團的覆蓋區間
3 段區間即可覆蓋 [0,M] 區間
1
2
3
覆蓋區間
4
5
6
覆蓋區間
maxl
7
8
9
左邊
覆蓋區間

- 何時有解？

  即可以找到最少的區間來覆蓋 [0,M] 區間

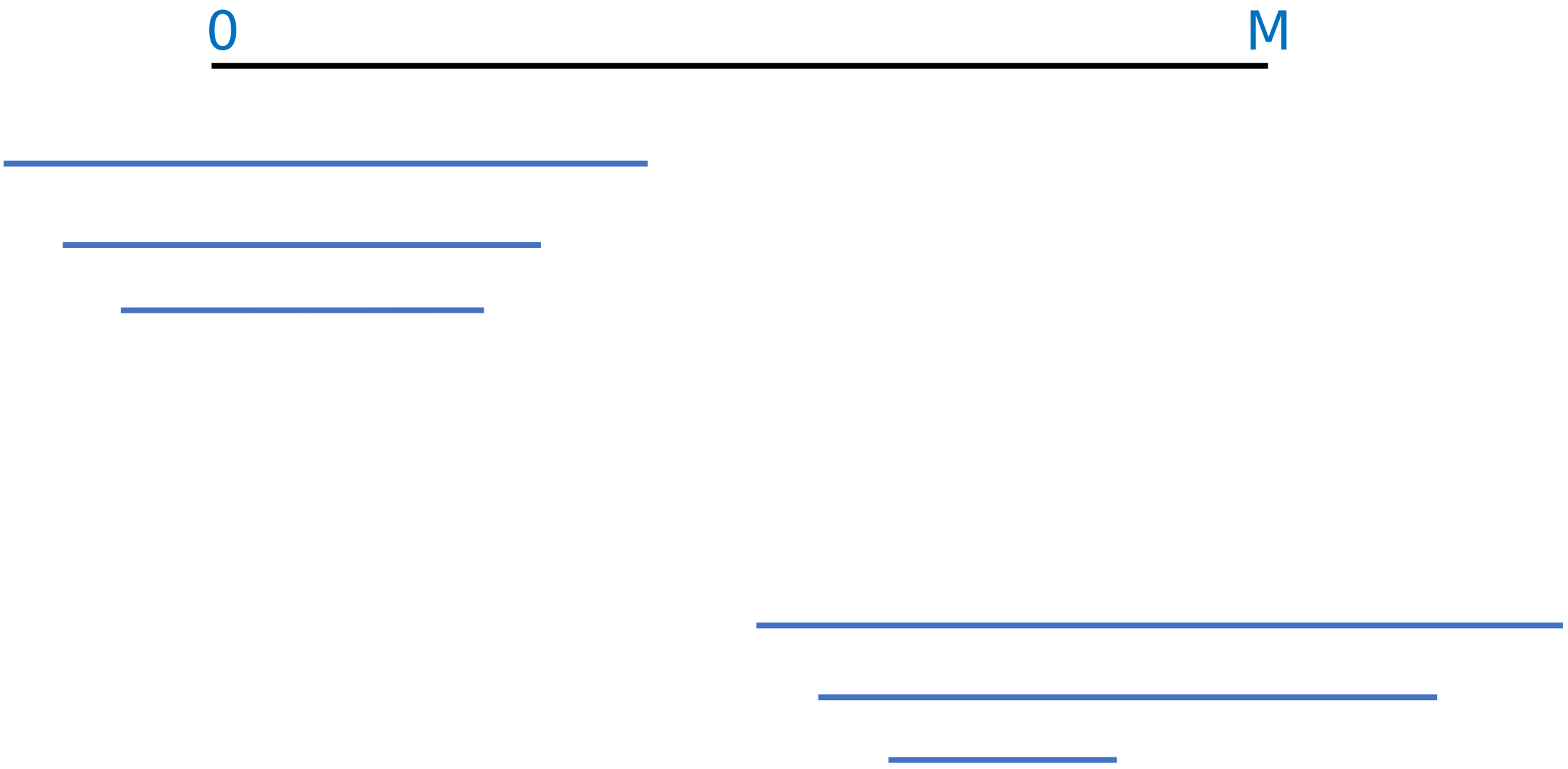
  - 當由左至右掃描左端點，到最後發現最後一團之 N.right > M 表示有解，而最少的區間數 (覆蓋 [0,M]) 就是團數 。
  - 如果最後一團之 N.left=N.right > M 表示有解，而最少的區間 (覆蓋 [0,M]) 不含這一團區間 。

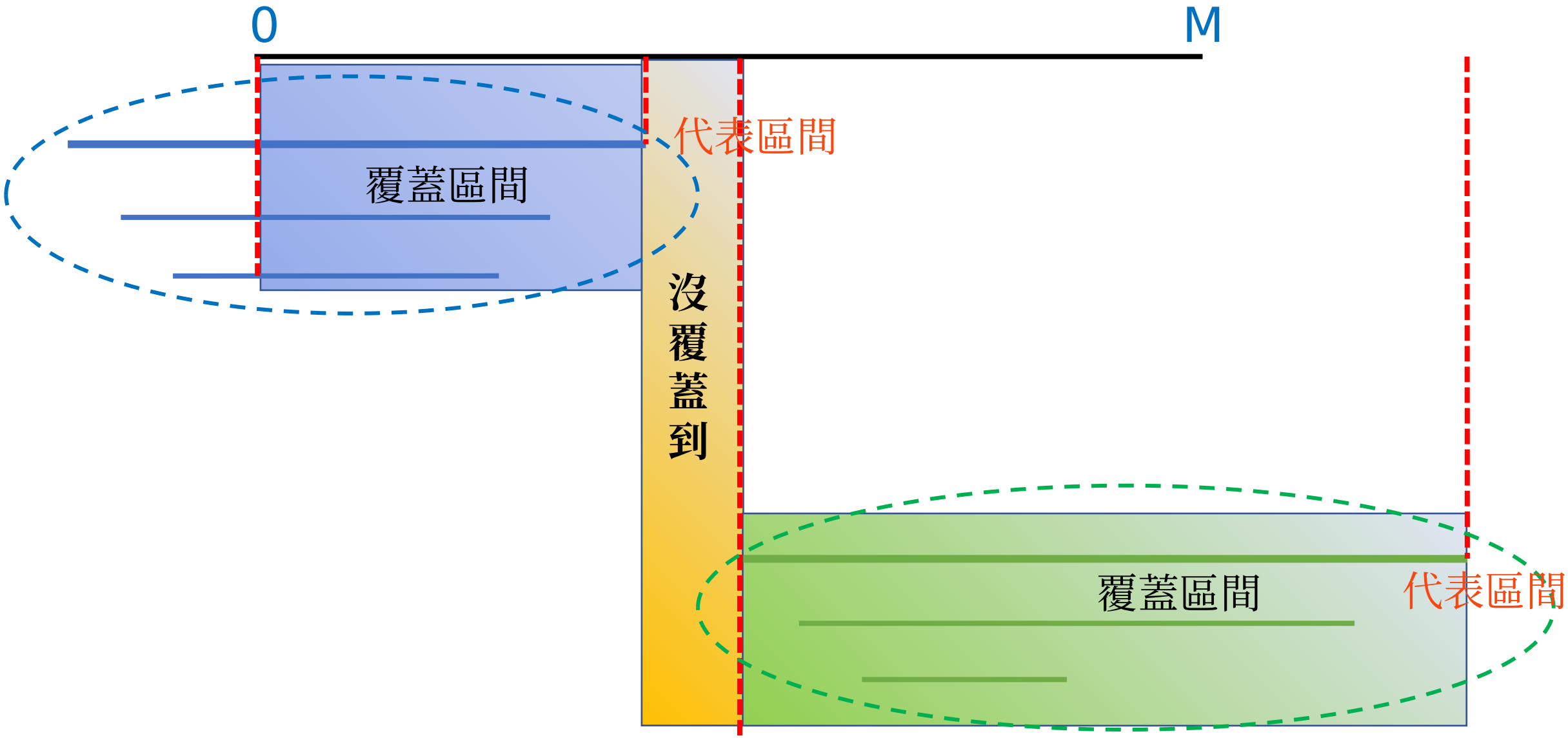
- 何時無解？

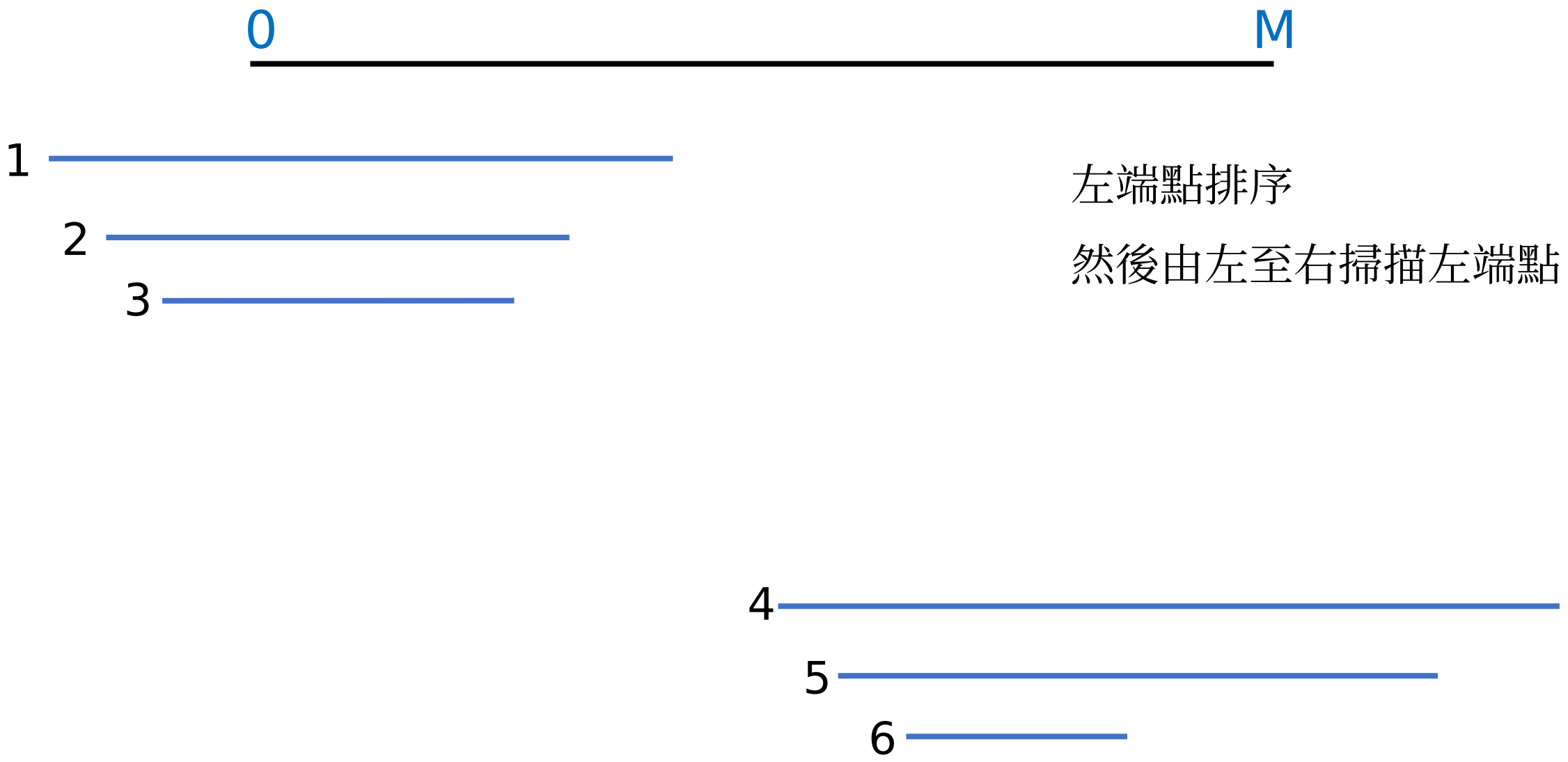
  即找不到區間來覆蓋 [0,M] 區間

  - 當由左至右掃描左端點，當中找到某團 N.left = N.right 而且 N.right < M 表示無解。
  - 掃描到最後一團之 N.right > M 表示有無解。

無解狀況

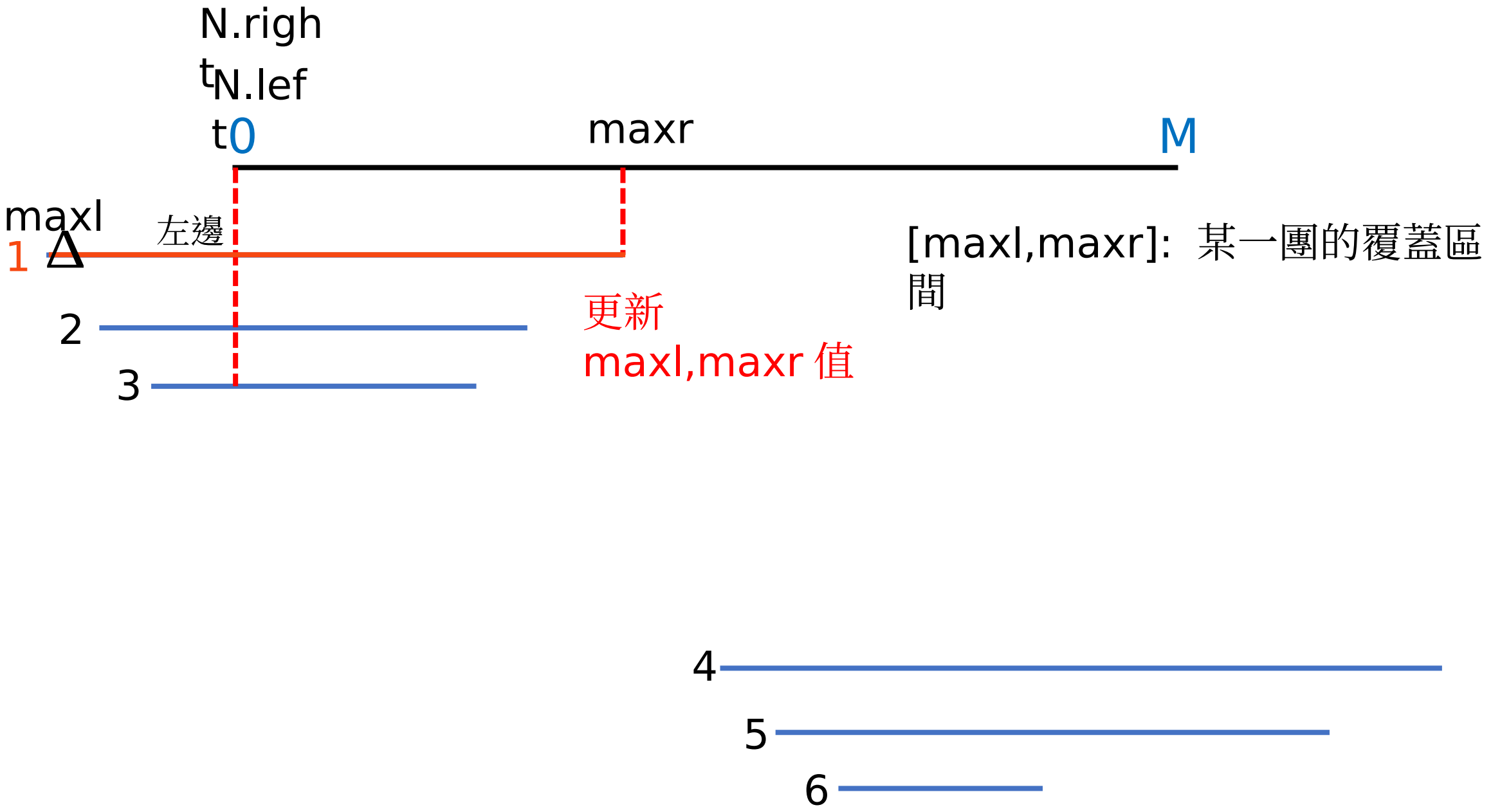

0
M

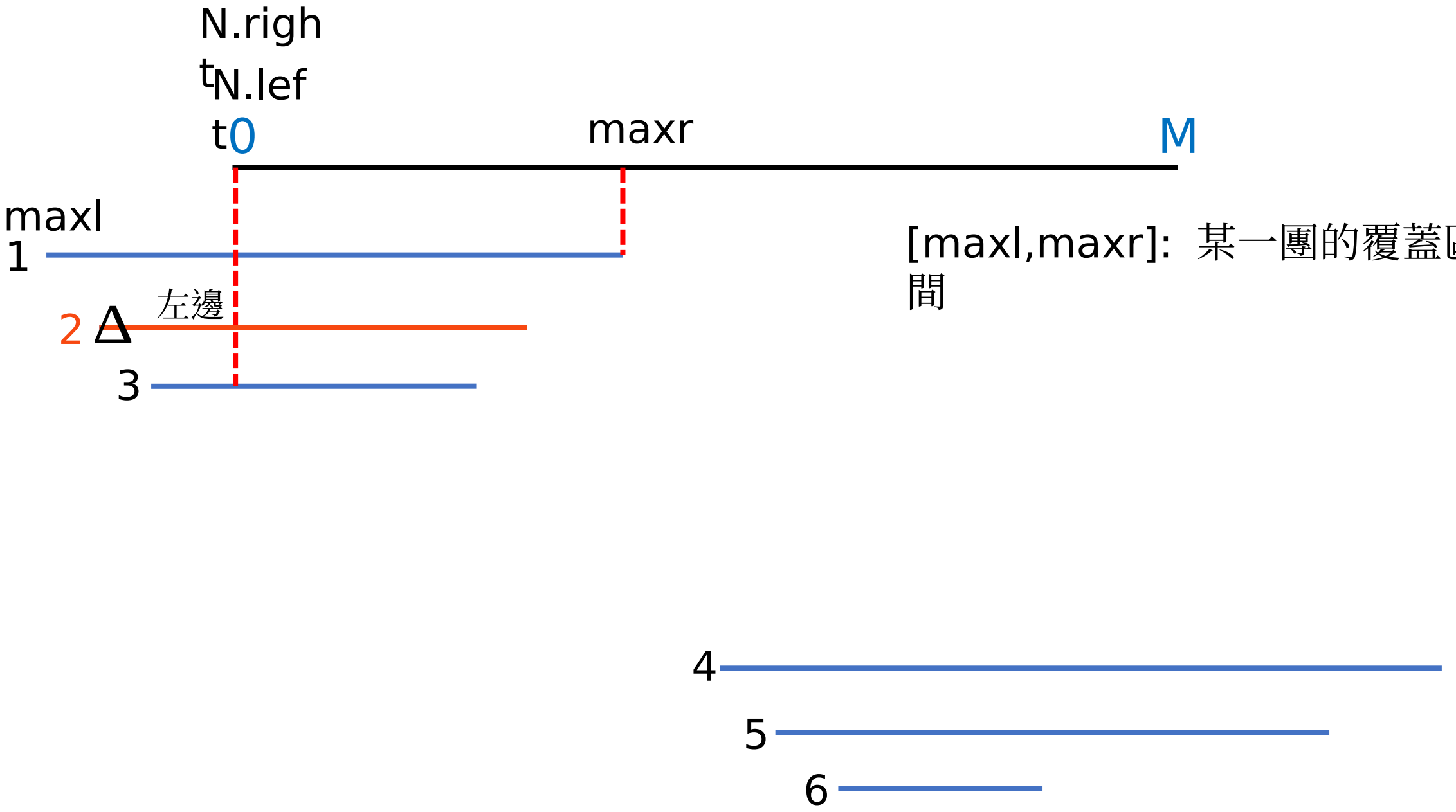

0
M
覆蓋區間
代表區間
沒覆蓋到
覆蓋區間
代表區間

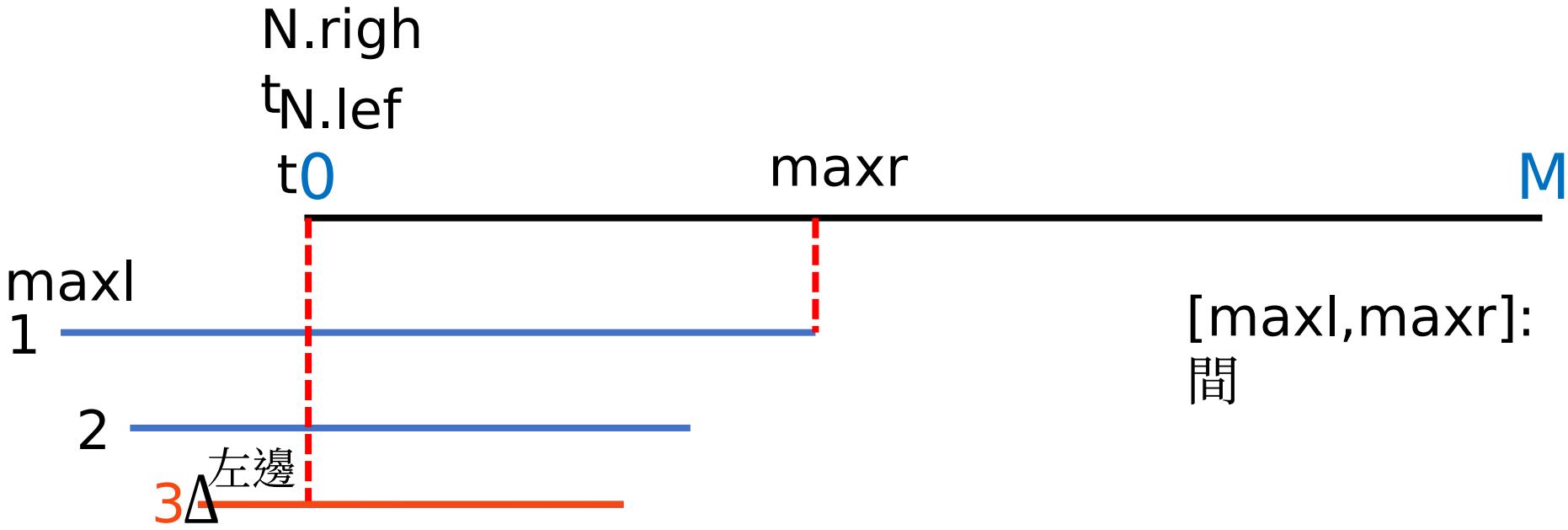


左端點排序

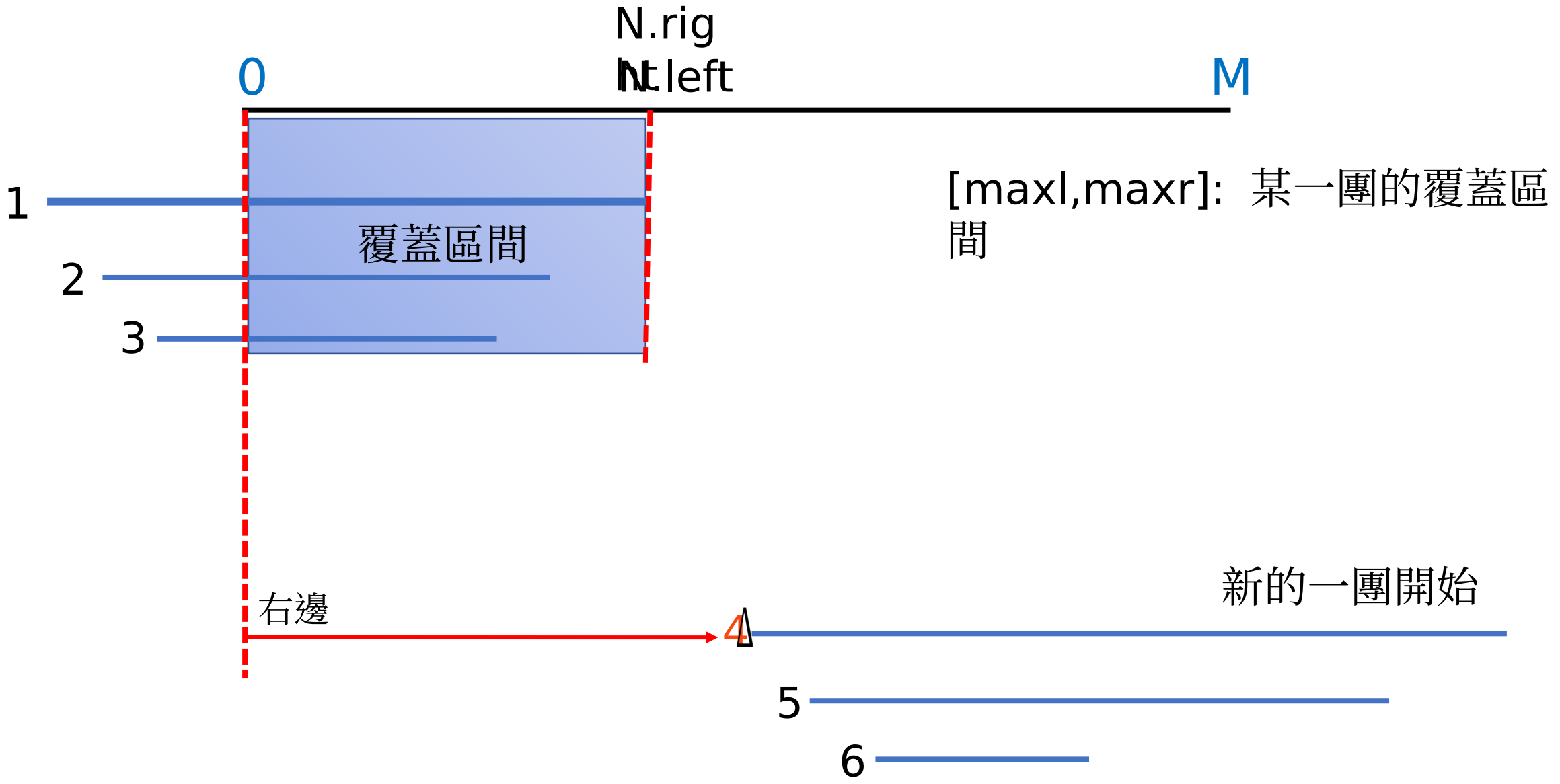
然後由左至右掃描左端點

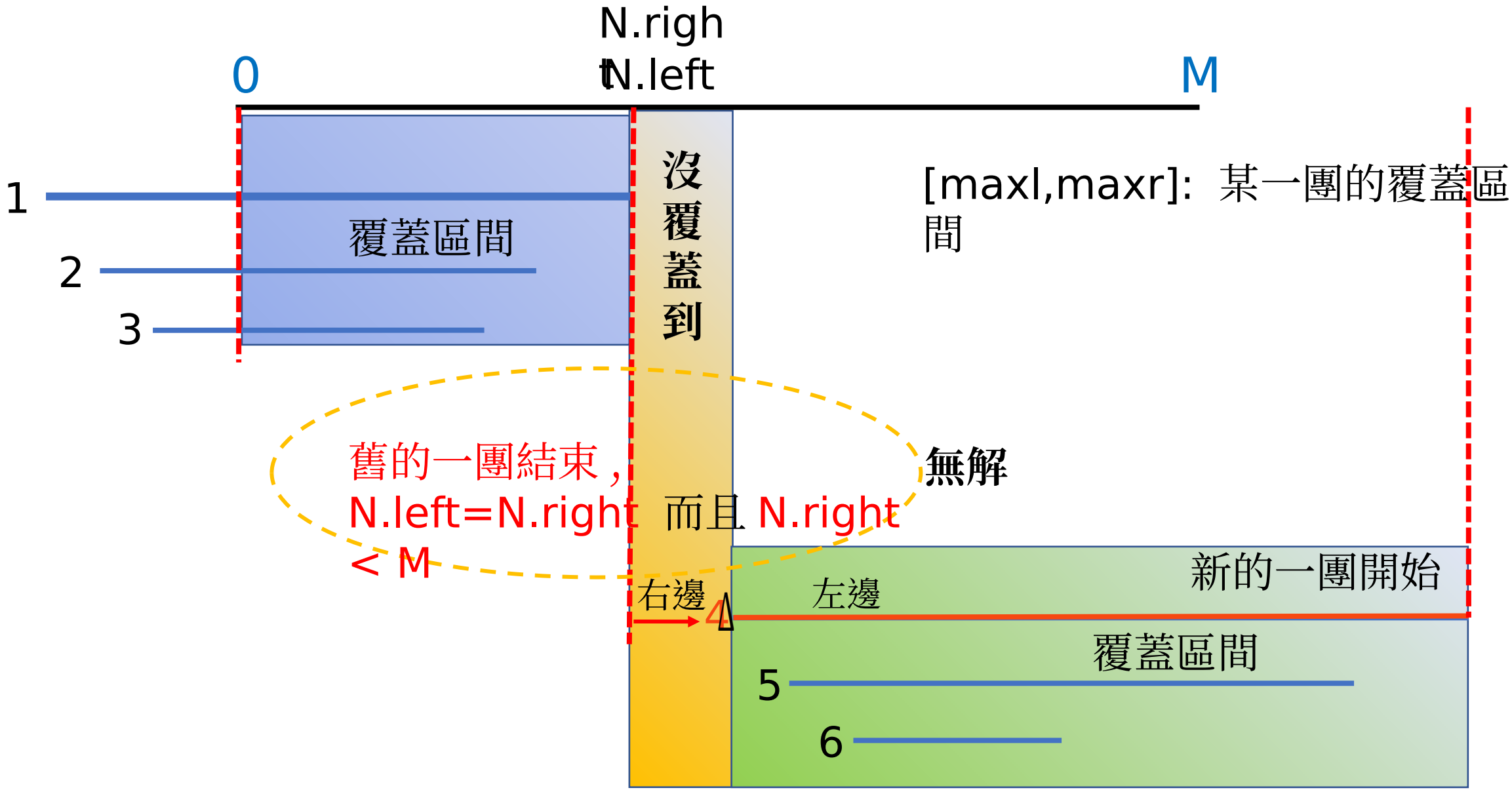

N.righ
t
N.lef
t
0
maxr
M
maxl
左邊
1
2
3
更新
maxl,maxr 值
[maxl,maxr]: 某一團的覆蓋區間
4
5
6


N.righ
t
N.lef
t
0
maxr
M
maxl
1
左邊
2
Δ
3
[maxl,maxr]: 某一團的覆蓋區間
4
5
6



[maxl,maxr]: 某一團的覆蓋區間


N.rig
N.left
0
M
1
2
3
覆蓋區間
[maxl,maxr]: 某一團的覆蓋區間
右邊
新的一團開始
4
5
6


N.righ
t
N.left
0
M
1
2
3
覆蓋區間
沒覆蓋到
[maxl,maxr]: 某一團的覆蓋區間
舊的一團結束，
N.left=N.right 而且 N.right < M
無解
右邊
左邊
新的一團開始
覆蓋區間
5
6

UVa 10020 Code (1/4)

```
#include<iostream>
#include<algorithm>
using namespace std;
struct line {
   int left,right;
 } k[100000], map[100000];     //  儲存每個區間左右端點值

bool cmp(line a,line b) {
    if (a.left==b.left)   return a.right > b.right;
      else return a.left < b.left;
}
```

UVa 10020 Code (2/4)

```
int main()
{
    int T,caseno;
    freopen("10020.in","r",stdin);
    freopen("10020.out","w",stdout);
    cin>>T; caseno=T;
    while(T--) {
        int m;
        cin>>m;
        int l,r,ans=0;
        while(cin>>l>>r) {
     if (l==0 && r==0)  break;
     k[ans].left=l;
     k[ans++].right=r;
          }
          sort(k,k+ans,cmp);  // ans 個區間依左端點由小至大排序
```

UVa 10020 Code (3/4)

```
    line N;
    N.left=0;   N.right=0; // [N.left,N.right]:  每一團覆蓋的範圍 , 第一團範圍 : [0,0]
    int maxr=0,maxl;    //  [maxl,maxr] 儲存覆蓋的區間
    int num=0;          //  num: 覆蓋的區間數
    int flag, i;
            i=0;
    while(N.left<m) {  //  掃描左端點由左至右
        flag=0;
        while (i<ans && k[i].left<=N.left) {                //  第 i 區間左端點在 N.left 左邊
                    if (k[i].right>N.right && k[i].right>maxr) {  //  更新 right 值 ( 向右移 )
                maxr=k[i].right;
                maxl=k[i].left;
                flag=1;
            }
                    i++;
        }   // end of while (i<ans...)
        if (flag==0)  break;   // flag=0:  無解
        N.left=maxr;          //  下一團開始範圍 [N.left,N.right], N.left=N.right
        N.right=maxr;
```

UVa 10020 Code (4/4)

```
            // 輸出
        if (T!=caseno-1) cout<<endl;
        if(flag==0)  cout<<"0"<<endl<<endl;
            else {
              cout<<num<<endl;
              for (int i=0; i<num; i++)
                 cout<<map[i].left<<" "<<map[i].right<<endl;
            }
     }  // end-of-while (T--)
}  // end
```