**线段树专辑**

这几天陆陆续续更新了下边几道我所能找到得具有一些代表性的线段树题目  
从最最简单的区间求和到对区间的各种操作都包涵在这些题目里了  
相信对一些准备学习线段树的人有一定得帮助  
突然发现自己对数据结构的题目非常有感觉,所以在刷下边的题的同时也生出灵感出了好几道线段树题目  
等比赛结束后也会陆续加进里边

**快半年过去代码风格也有很大的改变，感觉以前写的代码很不规范，用自己在预定义中定义的一些函数，但后来感觉作用不是很大，所以又删去了，所以现在看代码可能找不到以前我定义的一些函数，本来想更新一下的，无奈这些函数用的太多，改之太过麻烦，所以在此处申明一下  
#define LL(x) ((x)<<1)  
#define RR(x) ((x)<<1|1)  
#define FF(i,n) for(int i = 0 ; i < n ; i ++)  
若还有不清楚的地方，只管提出来便是，我一定一一改正**

[1.hdu1166 敌兵布阵](http://acm.hdu.edu.cn/showproblem.php?pid=1166)  
更新节点,区间求和

[?](http://www.ericbess.com/ericblog/2008/03/03/wp-codebox/#examples)[[Copy to clipboard]View Code](javascript:;) CPP

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75 | struct Seg\_Tree{  int left,right,num;  int calmid() {  return (left+right)/2;  }  }tt[150000];  int num[50001];  int build(int left,int right,int idx) {  tt[idx].left = left;  tt[idx].right = right;  if(left == right) {  return tt[idx].num = num[left];  }  int mid = (left + right)/2;  return tt[idx].num = build(left,mid,LL(idx)) + build(mid+1,right,RR(idx));  }    void update(int id,int x,int idx) {  tt[idx].num += x;  if(tt[idx].left == tt[idx].right) {  return ;  }  int mid = tt[idx].calmid();  if(id <= mid) {  update(id,x,LL(idx));  } else {  update(id,x,RR(idx));  }  }    int query(int left,int right,int idx) {  if(left == tt[idx].left && right == tt[idx].right) {  return tt[idx].num;  }  int mid = tt[idx].calmid();  if(right <= mid) {  return query(left,right,LL(idx));  } else if(mid < left) {  return query(left,right,RR(idx));  } else {  return query(left,mid,LL(idx)) + query(mid+1,right,RR(idx));  }  }    int main() {  int T;  scanf(“%d”,T);  FF(cas,T) {  int n;  scanf(“%d”,&n);  FOR(i,1,1+n) {  scanf(“%d”,num[i]);  }  build(1,n,1);  printf("Case %d:**\n**",cas+1);  char com[9];  while(scanf("%s",com)) {  if(strcmp(com,"End") == 0) break;  int a,b;  scanf("%d%d",&a,&b);  switch(com[0]) {  case 'Q':  printf("%d**\n**",query(a,b,1));  break;  case 'A':  update(a,b,1);  break;  case 'S':  update(a,-b,1);  break;  }  }  }  return 0;  } |

.  
[2.hdu1754 I Hate It](http://acm.hdu.edu.cn/showproblem.php?pid=1754)  
更新节点,区间最值

[?](http://www.ericbess.com/ericblog/2008/03/03/wp-codebox/#examples)[[Copy to clipboard]View Code](javascript:;) CPP

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66 | struct Seg\_Tree {  int left,right,val;  int calmid() {  return (left+right)/2;  }  }tt[600000];  int val[200001];    int build(int left,int right,int idx) {  tt[idx].left = left;  tt[idx].right = right;  if(left == right) {  return tt[idx].val = val[left];  }  int mid = tt[idx].calmid();  return tt[idx].val = max(build(left,mid,LL(idx)) , build(mid+1,right,RR(idx)));  }    void update(int id,int x,int idx) {  if(tt[idx].left == tt[idx].right){  tt[idx].val = x;  return ;  }  int mid = tt[idx].calmid();  if(id <= mid) {  update(id,x,LL(idx));  } else {  update(id,x,RR(idx));  }  tt[idx].val = max(tt[LL(idx)].val , tt[RR(idx)].val);  }    int query(int left,int right,int idx) {  if(left == tt[idx].left && right == tt[idx].right) {  return tt[idx].val;  }  int mid = tt[idx].calmid();  if(right <= mid) {  return query(left,right,LL(idx));  } else if(mid < left) {  return query(left,right,RR(idx));  } else {  return max(query(left,mid,LL(idx)) , query(mid+1,right,RR(idx)));  }  }    int main() {  int n , m ;  while(scanf("%d%d",&n,&m) == 2) {  FOR(i,1,n+1) {  scanf("%d",&val[i]);  }  build(1,n,1);  while(m --) {  char com[2];  int a,b;  scanf("%s%d%d",com,&a,&b);  if(com[0] == 'Q') {  printf("%d**\n**",query(a,b,1));  } else {  update(a,b,1);  }  }  }  return 0;  } |

.  
[3.hdu1698 Just a Hook](http://acm.hdu.edu.cn/showproblem.php?pid=1698)  
成段更新,总区间求和

[?](http://www.ericbess.com/ericblog/2008/03/03/wp-codebox/#examples)[[Copy to clipboard]View Code](javascript:;) CPP

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66 | int n;  struct Seg\_Tree{  int left,right,col;  }tt[300000];    void build(int left,int right,int idx) {  tt[idx].left = left;  tt[idx].right = right;  tt[idx].col = 1;  if(left == right) return ;  int mid = (left + right)/2;  build(left,mid,LL(idx));  build(mid+1,right,RR(idx));  }    void update(int left,int right,int col,int idx) {  if(left <= tt[idx].left && right >= tt[idx].right) {  tt[idx].col = col;  return ;  }  if(tt[idx].col != -1) {  tt[LL(idx)].col = tt[RR(idx)].col = tt[idx].col;  tt[idx].col = -1;  }  int mid = (tt[idx].left + tt[idx].right)/2;  if(left <= mid) update(left,right,col,LL(idx));  if(mid < right) update(left,right,col,RR(idx));    }    int query(int left,int right,int idx) {  if(left == tt[idx].left && right == tt[idx].right) {  if(tt[idx].col != -1) {  return (right - left + 1) \* tt[idx].col;  } else {  int mid = (tt[idx].left + tt[idx].right)/2;  return query(left,mid,LL(idx)) + query(mid+1,right,RR(idx));  }  }  int mid = (tt[idx].left + tt[idx].right)/2;  if(right <= mid) {  return query(left,right,LL(idx));  } else if(left > mid) {  return query(left,right,RR(idx));  } else {  return query(left,mid,LL(idx)) + query(mid+1,right,RR(idx));  }  }    int main() {  int T;  scanf("%d",&T);  FF(cas,T) {  int n , m ;  scanf("%d",&n);  build(1,n,1);  scanf("%d",&m);  while(m --) {  int left,right,col;  scanf("%d%d%d",&left,&right,&col);  update(left,right,col,1);  }  printf("Case %d: The total value of the hook is %d.**\n**",cas+1,query(1,n,1));  }  return 0;  } |

.  
[4.hdu1394 Minimum Inversion Number](http://acm.hdu.edu.cn/showproblem.php?pid=1394)  
更新节点,区间求和(实现nlog(n)的逆序数方法,和树状数组比起来实在是有点鸡肋)

[?](http://www.ericbess.com/ericblog/2008/03/03/wp-codebox/#examples)[[Copy to clipboard]View Code](javascript:;) CPP

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66 | struct Seg\_Tree {  int left,right,val;  int calmid() {  return (left+right)/2;  }  }tt[15000];  int val[5001];    void build(int left,int right,int idx) {  tt[idx].left = left;  tt[idx].right = right;  tt[idx].val = 0;  if(left == right) {  return ;  }  int mid = tt[idx].calmid();  build(left,mid,LL(idx));  build(mid+1,right,RR(idx));  }    int query(int left,int right,int idx) {  if(left == tt[idx].left && right == tt[idx].right) {  return tt[idx].val;  }  int mid = tt[idx].calmid();  if(right <= mid) {  return query(left,right,LL(idx));  } else if(mid < left) {  return query(left,right,RR(idx));  } else {  return query(left,mid,LL(idx)) + query(mid+1,right,RR(idx));  }  }    void update(int id,int idx) {  tt[idx].val ++;  if(tt[idx].left == tt[idx].right) {  return ;  }  int mid = tt[idx].calmid();  if(id <= mid) {  update(id,LL(idx));  } else {  update(id,RR(idx));  }  }    int main() {  int n;  while(scanf("%d",&n) == 1) {  build(0,n-1,1);  int sum = 0;  FF(i,n) {  scanf(“%d”,&val[i]);  sum += query(val[i],n-1,1);  update(val[i],1);  }  int ret = sum;  FF(i,n) {  sum = sum - val[i] + (n - val[i] - 1);  checkmin(ret,sum);  }  printf("%d**\n**",ret);  }  return 0;  } |

.  
[5.hdu1779 9-Problem C](http://acm.hdu.edu.cn/showproblem.php?pid=1779)暂不公开  
成段更新,区间最值

[?](http://www.ericbess.com/ericblog/2008/03/03/wp-codebox/#examples)[[Copy to clipboard]View Code](javascript:;) CPP

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96 | struct Node{  int idx;  int val;  friend bool operator < (Node a,Node b) {  if(a.val == b.val) {  return a.idx > b.idx;  }  return a.val < b.val;  }  };  struct Seg\_Tree {  int left,right;  Node node;  int add;  int calmid() {  return (left+right)/2;  }  }tt[300000];    void build(int left,int right,int idx) {  tt[idx].left = left;  tt[idx].right = right;  tt[idx].add = 0;  tt[idx].node.idx = left;  tt[idx].node.val = 0;  if(left == right) {  return ;  }  int mid = tt[idx].calmid();  build(left,mid,LL(idx));  build(mid+1,right,RR(idx));  }    void update(int left,int right,int add,int idx) {  if(left <= tt[idx].left && right >= tt[idx].right) {  tt[idx].node.val += add;  tt[idx].add += add;  return ;  }  if(tt[idx].add) {  tt[LL(idx)].node.val += tt[idx].add;  tt[RR(idx)].node.val += tt[idx].add;  tt[LL(idx)].add += tt[idx].add;  tt[RR(idx)].add += tt[idx].add;  tt[idx].add = 0;  }  int mid = tt[idx].calmid();  if(left <= mid) update(left,right,add,LL(idx));  if(mid < right) update(left,right,add,RR(idx));  tt[idx].node = max(tt[LL(idx)].node , tt[RR(idx)].node);  }    Node query(int left,int right,int idx) {  if(left == tt[idx].left && right == tt[idx].right) {  return tt[idx].node;  }  if(tt[idx].add) {  tt[LL(idx)].node.val += tt[idx].add;  tt[RR(idx)].node.val += tt[idx].add;  tt[LL(idx)].add += tt[idx].add;  tt[RR(idx)].add += tt[idx].add;  tt[idx].add = 0;  }  int mid = tt[idx].calmid();  if(right <= mid) {  return query(left,right,LL(idx));  } else if(left > mid) {  return query(left,right,RR(idx));  } else {  return max(query(left,mid,LL(idx)) , query(mid+1,right,RR(idx)));  }  }    int main() {  int n , m ;  while(scanf("%d%d",&n,&m) == 2) {  if(n == 0 && m == 0) break;  build(1,n,1);  while(m --) {  char com[2];  scanf("%s",&com);  if(com[0] == 'I') {  int a,b,c;  scanf("%d%d%d",&a,&b,&c);  update(a,b,c,1);  } else {  int a,b;  scanf("%d%d",&a,&b);  Node ret = query(a,b,1);  printf("%d**\n**",ret.val);  update(ret.idx,ret.idx,-ret.val,1);  }  }  }  return 0;  } |

.  
[6.pku2777 Count Color](http://acm.pku.edu.cn/JudgeOnline/problem?id=2777)  
成段更新,区间统计,位运算加速(我总把query里的传递给子节点的步骤忘了)

[?](http://www.ericbess.com/ericblog/2008/03/03/wp-codebox/#examples)[[Copy to clipboard]View Code](javascript:;) CPP

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82 | template<class T> inline T countbit(T n) {return n?1+countbit(n&(n-1)):0;}  #define two(x) ((LL)1<<(x))  struct Seg\_Tree{  int left,right;  int col;  bool cover;  int calmid() {  return (left+right)/2;  }  }tt[300000];    void build(int left,int right,int idx) {  tt[idx].left = left;  tt[idx].right = right;  tt[idx].col = two(1);  tt[idx].cover = true;  if(left == right) {  return ;  }  int mid = tt[idx].calmid();  build(left,mid,LL(idx));  build(mid+1,right,RR(idx));  }    void update(int left,int right,int c,int idx) {  if(left <= tt[idx].left && right >= tt[idx].right) {  tt[idx].cover = true;  tt[idx].col = two(c);  return ;  }  if(tt[idx].cover) {  tt[RR(idx)].col = tt[LL(idx)].col = tt[idx].col;  tt[RR(idx)].cover = tt[LL(idx)].cover = true;  tt[idx].cover = false;  }  int mid = tt[idx].calmid();  if(left <= mid) update(left,right,c,LL(idx));  if(mid < right) update(left,right,c,RR(idx));  tt[idx].col = tt[LL(idx)].col | tt[RR(idx)].col;  }    int query(int left,int right,int idx) {  if(left == tt[idx].left && right == tt[idx].right) {  return tt[idx].col;  }  if(tt[idx].cover) {  tt[RR(idx)].col = tt[LL(idx)].col = tt[idx].col;  tt[RR(idx)].cover = tt[LL(idx)].cover = true;  tt[idx].cover = false;  }  int mid = tt[idx].calmid();  if(right <= mid) {  return query(left,right,LL(idx));  } else if(mid < left) {  return query(left,right,RR(idx));  } else {  return query(left,mid,LL(idx)) | query(mid+1,right,RR(idx));  }  }    int main() {  int n,T,o;  while(scanf("%d%d%d",&n,&T,&o) == 3) {  build(1,n,1);  while(o--) {  char ch[2];  scanf("%s",&ch);  if(ch[0] == 'C') {  int a,b,c;  scanf("%d%d%d",&a,&b,&c);  if(a > b) swap(a,b);  update(a,b,c,1);  } else {  int a,b;  scanf("%d%d",&a,&b);  if(a > b) swap(a,b);  printf("%d**\n**",countbit(query(a,b,1)));  }  }  }  return 0;  } |

.  
[7.pku3468 A Simple Problem with Integers](http://acm.pku.edu.cn/JudgeOnline/problem?id=3468)  
成段更新,区间求和(中间乘法会超int..纠结了)

[?](http://www.ericbess.com/ericblog/2008/03/03/wp-codebox/#examples)[[Copy to clipboard]View Code](javascript:;) CPP

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88 | struct Seg\_Tree{  int left,right;  LL sum;  int add;  int calmid() {  return (left+right)/2;  }  int caldis() {  return right - left + 1;  }  }tt[400000];    int val[100001];    LL build(int l,int r,int idx) {  tt[idx].left = l;  tt[idx].right = r;  tt[idx].add = 0;  if(l == r) {  return tt[idx].sum = val[l];  }  int mid = tt[idx].calmid();  return tt[idx].sum = build(l,mid,LL(idx)) + build(mid+1,r,RR(idx));  }    void update(int l,int r,int add,int idx) {  if(l <= tt[idx].left && r >= tt[idx].right) {  tt[idx].add += add;  tt[idx].sum += (LL)tt[idx].caldis() \* add;  return ;  }  if(tt[idx].add) {  tt[LL(idx)].sum += (LL)tt[LL(idx)].caldis() \* tt[idx].add;  tt[RR(idx)].sum += (LL)tt[RR(idx)].caldis() \* tt[idx].add;  tt[LL(idx)].add += tt[idx].add;  tt[RR(idx)].add += tt[idx].add;  tt[idx].add = 0;  }  int mid = tt[idx].calmid();  if(l <= mid) update(l,r,add,LL(idx));  if(mid < r) update(l,r,add,RR(idx));  tt[idx].sum = tt[LL(idx)].sum + tt[RR(idx)].sum;  }    LL query(int l,int r,int idx) {  if(l == tt[idx].left && r == tt[idx].right) {  return tt[idx].sum;  }  if(tt[idx].add) {  tt[LL(idx)].sum += (LL)tt[LL(idx)].caldis() \* tt[idx].add;  tt[RR(idx)].sum += (LL)tt[RR(idx)].caldis() \* tt[idx].add;  tt[LL(idx)].add += tt[idx].add;  tt[RR(idx)].add += tt[idx].add;  tt[idx].add = 0;  }  int mid = tt[idx].calmid();  if(r <= mid) {  return query(l,r,LL(idx));  } else if(mid < l) {  return query(l,r,RR(idx));  } else {  return query(l,mid,LL(idx)) + query(mid+1,r,RR(idx));  }  }    int main() {  int n , m ;  while(scanf("%d%d",&n,&m) == 2) {  FOR(i,1,n+1) {  scanf("%d",&val[i]);  }  build(1,n,1);  while(m --) {  char com[2];  scanf("%s",com);  if(com[0] == 'Q') {  int a,b;  scanf("%d%d",&a,&b);  printf("%lld**\n**",query(a,b,1));  } else {  int a,b,c;  scanf("%d%d%d",&a,&b,&c);  update(a,b,c,1);  }  }  }  return 0;  } |

.  
[8.pku2528 Mayor’s posters](http://acm.pku.edu.cn/JudgeOnline/problem?id=2528)  
成段更新,区间统计(离散化)

[?](http://www.ericbess.com/ericblog/2008/03/03/wp-codebox/#examples)[[Copy to clipboard]View Code](javascript:;) CPP

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104 | struct Seg\_Tree{  int left;  int right;  int col;  int mid() {  return (left+right)>>1;  }  }tt[66666];    struct H{  int left,right;  }hh[10001];  int pos[20001];  bool hash[10001];  int n , m ,ans ;    void build(int l,int r,int idx) {  tt[idx].left = l;  tt[idx].right = r;  tt[idx].col = -1;  if(l == r) return ;  int mid = tt[idx].mid();  build(l,mid,LL(idx));  build(mid+1,r,RR(idx));  }    void update(int l,int r,int col,int idx) {  if(l <= tt[idx].left && r >= tt[idx].right) {  tt[idx].col = col;  return ;  }  if(tt[idx].col != -1) {  tt[LL(idx)].col = tt[RR(idx)].col = tt[idx].col;  tt[idx].col = -1;  }  int mid = tt[idx].mid();  if(l <= mid) update(l,r,col,LL(idx));  if(mid < r) update(l,r,col,RR(idx));  }    void query(int l,int r,int idx) {  if(tt[idx].col != -1) {  if(!hash[tt[idx].col]) {  ans ++;  hash[tt[idx].col] = true;  }  return ;  }  int mid = tt[idx].mid();  if(r <= mid) {  query(l,r,LL(idx));  } else if(mid < l) {  query(l,r,RR(idx));  } else {  query(l,mid,LL(idx));  query(mid+1,r,RR(idx));  }  }    int Bin(int x) {  int lo = 0;  int hi = m-1;  while(lo <= hi) {  int mid = (lo + hi) >> 1;  if(pos[mid] == x) {  return mid;  }  if(pos[mid] < x) {  lo = mid + 1;  } else {  hi = mid - 1;  }  }  return -1;  }    int main() {  int T;  scanf("%d",&T);  while(T --) {  scanf("%d",&n);  FF(i,n) {  scanf("%d%d",&hh[i].left,&hh[i].right);  pos[LL(i)] = hh[i].left;  pos[RR(i)] = hh[i].right;  }  sort(pos,pos+2\*n);  m = 0;  FF(i,2\*n) {  if(i == 0 || pos[i] != pos[i-1]) {  pos[m++] = pos[i];  }  }  build(0,m-1,1);  FF(i,n) {  update(Bin(hh[i].left),Bin(hh[i].right),i,1);  }  ans = 0;  CC(hash,false);  query(0,m-1,1);  printf("%d**\n**",ans);  }  return 0;  } |

.  
[9.hdu2795 Billboard](http://acm.hdu.edu.cn/showproblem.php?pid=2795)  
更新节点,询问特殊(暑假的时候不会线段树被这水题狂虐)

[?](http://www.ericbess.com/ericblog/2008/03/03/wp-codebox/#examples)[[Copy to clipboard]View Code](javascript:;) CPP

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74 | struct Node {  int val;  int idx;  friend bool operator < (Node a , Node b) {  if(a.val == b.val) {  return a.idx > b.idx;  }  return a.val < b.val;  }  }error;    struct Seg\_Tree{  int left,right;  Node node;  int mid() {  return (left + right)>>1;  }  }tt[800000];    int n , h , m;    void build(int l,int r,int idx) {  tt[idx].left = l;  tt[idx].right = r;  tt[idx].node.idx = l;  tt[idx].node.val = h;  if(l == r) return ;  int mid = tt[idx].mid();  build(l,mid,LL(idx));  build(mid+1,r,RR(idx));  }    void update(int l,int r,int val,int idx) {  if(l <= tt[idx].left && r >= tt[idx].right) {  tt[idx].node.val += val;  return ;  }  int mid = tt[idx].mid();  if(l <= mid) update(l,r,val,LL(idx));  if(mid < r) update(l,r,val,RR(idx));  tt[idx].node = max(tt[LL(idx)].node,tt[RR(idx)].node);  }    Node query(int w,int idx) {  if(tt[idx].node.val < w) {  return error;  }  if(tt[idx].left == tt[idx].right) {  return tt[idx].node;  }  if(tt[LL(idx)].node.val >= w) {  return query(w,LL(idx));  } else {  return query(w,RR(idx));  }  }    int main() {  error.idx = -1;  while(scanf("%d%d%d",&n,&h,&m) == 3) {  checkmin(n,m);  build(1,n,1);  while(m --) {  int w;  scanf("%d",&w);  Node ret = query(w,1);  printf("%d**\n**",ret.idx);  if(ret.idx != -1) {  update(ret.idx,ret.idx,-w,1);  }  }  }  return 0;  } |

.  
[10.pku3667 Hotel](http://acm.pku.edu.cn/JudgeOnline/problem?id=3667)  
成段更新,寻找空间(经典类型,求一块满足条件的最左边的空间)

[?](http://www.ericbess.com/ericblog/2008/03/03/wp-codebox/#examples)[[Copy to clipboard]View Code](javascript:;) CPP

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98 | struct Seg\_Tree{  int left,right;  int cval,lval,rval;  int st;  int mid() {  return (left+right)>>1;  }  void doit() {  cval = lval = rval = st ? 0 : dis();  }  int dis() {  return (right - left + 1);  }  }tt[150000];    void build(int l,int r,int idx) {  tt[idx].left = l;  tt[idx].right = r;  tt[idx].st = -1;  tt[idx].cval = tt[idx].lval = tt[idx].rval = tt[idx].dis();  if(l == r) return ;  int mid = tt[idx].mid();  build(l,mid,LL(idx));  build(mid+1,r,RR(idx));  }    void update(int l,int r,int st,int idx) {  if(tt[idx].left >= l && r >= tt[idx].right) {  tt[idx].st = st;  tt[idx].doit();  return ;  }  if(tt[idx].st != -1) {  tt[LL(idx)].st = tt[RR(idx)].st = tt[idx].st;  tt[LL(idx)].doit();  tt[RR(idx)].doit();  tt[idx].st = -1;  }  int mid = tt[idx].mid();  if(l <= mid) update(l,r,st,LL(idx));  if(mid < r) update(l,r,st,RR(idx));    tt[idx].cval = max( tt[LL(idx)].rval + tt[RR(idx)].lval , max(tt[LL(idx)].cval,tt[RR(idx)].cval) );  tt[idx].lval = tt[LL(idx)].lval;  tt[idx].rval = tt[RR(idx)].rval;  if(tt[LL(idx)].cval == tt[LL(idx)].dis()) {  tt[idx].lval += tt[RR(idx)].lval;  }  if(tt[RR(idx)].cval == tt[RR(idx)].dis()) {  tt[idx].rval += tt[LL(idx)].rval;  }  }    int query(int w,int idx) {  if(tt[idx].left == tt[idx].right && w == 1) {  return tt[idx].left;  }  if(tt[idx].st != -1) {  tt[LL(idx)].st = tt[RR(idx)].st = tt[idx].st;  tt[LL(idx)].doit();  tt[RR(idx)].doit();  tt[idx].st = -1;  }  if(tt[LL(idx)].cval >= w) {  return query(w,LL(idx));  } else if(tt[LL(idx)].rval + tt[RR(idx)].lval >= w) {  return tt[LL(idx)].right - tt[LL(idx)].rval + 1;  } else if(tt[RR(idx)].cval >= w){  return query(w,RR(idx));  } else {  return 0;  }  }    int main() {  int n , m;  while(scanf("%d%d",&n,&m) == 2) {  build(1,n,1);  while(m --) {  int com;  scanf("%d",&com);  if(com == 1) {  int a;  scanf("%d",&a);  int s = query(a,1);  printf("%d**\n**",s);  if(s) {  update(s,s+a-1,1,1);  }  } else {  int a,b;  scanf("%d%d",&a,&b);  update(a,a+b-1,0,1);  }  }  }  return 0;  } |

.  
[11.hdu1540 Tunnel Warfare](http://acm.hdu.edu.cn/showproblem.php?pid=1540)  
更新节点,询问节点所在区间(同上一道Hotel一样类型的题目)

[?](http://www.ericbess.com/ericblog/2008/03/03/wp-codebox/#examples)[[Copy to clipboard]View Code](javascript:;) CPP

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94 | struct Seg\_Tree{  int left,right;  int cval,lval,rval;  int mid() {  return (left+right)>>1;  }  void doit(int st) {  cval = lval = rval = st ? 0 : dis();  }  int dis() {  return (right - left + 1);  }  }tt[150000];    void build(int l,int r,int idx) {  tt[idx].left = l;  tt[idx].right = r;  tt[idx].cval = tt[idx].lval = tt[idx].rval = tt[idx].dis();  if(l == r) return ;  int mid = tt[idx].mid();  build(l,mid,LL(idx));  build(mid+1,r,RR(idx));  }    void update(int id,int st,int idx) {  if(tt[idx].left == tt[idx].right) {  tt[idx].doit(st);  return ;  }  int mid = tt[idx].mid();  if(id <= mid) {  update(id,st,LL(idx));  } else {  update(id,st,RR(idx));  }  int l = LL(idx);  int r = RR(idx);  tt[idx].cval = max(tt[l].rval + tt[r].lval , max(tt[l].cval , tt[r].cval));  tt[idx].lval = tt[l].lval;  tt[idx].rval = tt[r].rval;  if(tt[l].cval == tt[l].dis()) {  tt[idx].lval += tt[r].lval;  }  if(tt[r].cval == tt[r].dis()) {  tt[idx].rval += tt[l].rval;  }  }    int query(int id,int idx) {  if(tt[idx].left == tt[idx].right || tt[idx].cval == 0 || tt[idx].cval == tt[idx].dis()) {  return tt[idx].cval;  }  int mid = tt[idx].mid();  if(id <= mid) {  if(id > tt[LL(idx)].right - tt[LL(idx)].rval) {  return query(id,LL(idx)) + query(tt[RR(idx)].left,RR(idx));  } else {  return query(id,LL(idx));  }  } else {  if(id < tt[RR(idx)].left + tt[RR(idx)].lval) {  return query(tt[LL(idx)].right,LL(idx)) + query(id,RR(idx));  } else {  return query(id,RR(idx));  }  }  }    stack <int> ss;  int main() {  int n , m ;  while(scanf("%d%d",&n,&m) == 2) {  build(1,n,1);  while(m --) {  char com[2];  scanf("%s",com);  if(com[0] == 'D') {  int idx;  scanf("%d",&idx);  ss.push(idx);  update(idx,1,1);  } else if(com[0] == 'Q') {  int idx;  scanf("%d",&idx);  printf("%d**\n**",query(idx,1));  } else {  if(ss.empty()) continue;  update(ss.top(),0,1);  ss.pop();  }  }  }  return 0;  } |

.  
[12.hdu2871 Memory Control](http://acm.hdu.edu.cn/showproblem.php?pid=2871)  
hotel变形题目,三个都函数一样(vector忘记清空检查了好久)

[?](http://www.ericbess.com/ericblog/2008/03/03/wp-codebox/#examples)[[Copy to clipboard]View Code](javascript:;) CPP

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122  123  124  125  126  127  128  129  130  131  132  133  134  135  136  137  138  139  140  141  142  143  144  145  146  147  148  149  150 | struct Seg\_Tree{  int left,right;  int cval,lval,rval;  int st;  int mid() {  return (left+right)>>1;  }  void doit() {  cval = lval = rval = st ? 0 : dis();  }  int dis() {  return (right - left + 1);  }  }tt[150000];    struct Block{  int start;  int end;  };  vector <Block> hh;    void build(int l,int r,int idx) {  tt[idx].left = l;  tt[idx].right = r;  tt[idx].cval = tt[idx].lval = tt[idx].rval = tt[idx].dis();  tt[idx].st = -1;  if(l == r) return ;  int mid = tt[idx].mid();  build(l,mid,LL(idx));  build(mid+1,r,RR(idx));  }    void update(int l,int r,int st,int idx) {  if(tt[idx].left >= l && r >= tt[idx].right) {  tt[idx].st = st;  tt[idx].doit();  return ;  }  if(tt[idx].st != -1) {  tt[LL(idx)].st = tt[RR(idx)].st = tt[idx].st;  tt[LL(idx)].doit();  tt[RR(idx)].doit();  tt[idx].st = -1;  }  int mid = tt[idx].mid();  if(l <= mid) update(l,r,st,LL(idx));  if(mid < r) update(l,r,st,RR(idx));    tt[idx].cval = max( tt[LL(idx)].rval + tt[RR(idx)].lval , max(tt[LL(idx)].cval,tt[RR(idx)].cval) );  tt[idx].lval = tt[LL(idx)].lval;  tt[idx].rval = tt[RR(idx)].rval;  if(tt[LL(idx)].cval == tt[LL(idx)].dis()) {  tt[idx].lval += tt[RR(idx)].lval;  }  if(tt[RR(idx)].cval == tt[RR(idx)].dis()) {  tt[idx].rval += tt[LL(idx)].rval;  }  }    int query(int w,int idx) {  if(tt[idx].left == tt[idx].right) {  if(tt[idx].cval == w) {  return tt[idx].left;  } else {  return 0;  }  }  if(tt[idx].st != -1) {  tt[LL(idx)].st = tt[RR(idx)].st = tt[idx].st;  tt[LL(idx)].doit();  tt[RR(idx)].doit();  tt[idx].st = -1;  }  if(tt[LL(idx)].cval >= w) {  return query(w,LL(idx));  } else if(tt[LL(idx)].rval + tt[RR(idx)].lval >= w) {  return tt[LL(idx)].right - tt[LL(idx)].rval + 1;  } else if(tt[RR(idx)].cval >= w){  return query(w,RR(idx));  } else {  return 0;  }  }    int Bin(int x) {  int lo = 0;  int hi = sz(hh) - 1;  while(lo <= hi) {  int mid = (lo + hi) >> 1;  if(hh[mid].start <= x) {  lo = mid + 1;  } else {  hi = mid - 1;  }  }  return lo;  }    int main() {  int n , m ;  while(scanf("%d%d",&n,&m) == 2) {  build(1,n,1);  hh.clear();  while(m --) {  char com[9];  scanf("%s",com);  if(strcmp(com,"Reset") == 0) {  update(1,n,0,1);  hh.clear();  puts("Reset Now");  } else if(strcmp(com,"Free") == 0) {  int idx;  scanf("%d",&idx);  int id = Bin(idx) - 1;  if(id == -1 || hh[id].end < idx) {  puts("Reject Free");  } else {  printf("Free from %d to %d**\n**",hh[id].start,hh[id].end);  update(hh[id].start,hh[id].end,0,1);  hh.erase(hh.begin()+id,hh.begin()+id+1);  }  } else if(strcmp(com,"New") == 0) {  int w;  scanf("%d",&w);  if(tt[1].cval < w) {  puts("Reject New");  } else {  Block buf;  buf.start = query(w,1);  buf.end = buf.start + w - 1;  int idx = Bin(buf.start);  hh.insert(hh.begin()+idx,buf);  update(buf.start,buf.end,1,1);  printf("New at %d**\n**",buf.start);  }  } else {  int idx;  scanf("%d",&idx);  idx --;  if(idx >= sz(hh)) {  puts("Reject Get");  } else {  printf("Get at %d**\n**",hh[idx].start);  }  }  }  puts("");  }  return 0;  } |

.  
[13.hdu3016 Man Down](http://acm.hdu.edu.cn/showproblem.php?pid=3016)  
成段更新,单点查询(简单线段树+简单DP)

[?](http://www.ericbess.com/ericblog/2008/03/03/wp-codebox/#examples)[[Copy to clipboard]View Code](javascript:;) CPP

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95 | #define mix -2    struct Seg\_Tree{  int left;  int right;  int id;  int mid() {  return (left + right)>>1;  }  }tt[400000];    struct H{  int left;  int right;  int xl,xr,val,h;  }hh[100001];  int n;  bool cmp(H a,H b) {  return a.h < b.h;  }    void build(int l,int r,int idx) {  tt[idx].left = l;  tt[idx].right = r;  tt[idx].id = -1;  if(l == r) return ;  int mid = tt[idx].mid();  build(l,mid,LL(idx));  build(mid+1,r,RR(idx));  }    void update(int l,int r,int id,int idx) {  if(l <= tt[idx].left && r >= tt[idx].right) {  tt[idx].id = id;  return ;  }  if(tt[idx].id != mix) {  tt[LL(idx)].id = tt[RR(idx)].id = tt[idx].id;  tt[idx].id = mix;  }  int mid = tt[idx].mid();  if(l <= mid) update(l,r,id,LL(idx));  if(mid < r) update(l,r,id,RR(idx));  }    int query(int pos,int idx) {  if(tt[idx].id != mix) {  return tt[idx].id;  }  if(tt[idx].id != mix) {  tt[LL(idx)].id = tt[RR(idx)].id = tt[idx].id;  tt[idx].id = mix;  }  int mid = tt[idx].mid();  if(pos <= mid) {  return query(pos,LL(idx));  } else {  return query(pos,RR(idx));  }  }    int dp[100001];    int main() {  while(scanf("%d",&n) == 1) {  FF(i,n) {  scanf("%d%d%d%d",&hh[i].h,&hh[i].xl,&hh[i].xr,&hh[i].val);  }  sort(hh,hh+n,cmp);  build(1,100000,1);  FF(i,n) {  hh[i].left = query(hh[i].xl,1);  hh[i].right = query(hh[i].xr,1);  update(hh[i].xl,hh[i].xr,i,1);  }  int ret = -1;  CC(dp,-1);  dp[n-1] = 100;  FFD(i,n) {  if(dp[i] == -1) continue;  if(hh[i].left == -1) {  checkmax(ret,dp[i] + hh[i].val);  } else {  checkmax(dp[hh[i].left] , dp[i] + hh[i].val);  }  if(hh[i].right == -1) {  checkmax(ret,dp[i] + hh[i].val);  } else {  checkmax(dp[hh[i].right] , dp[i] + hh[i].val);  }  }  printf("%d**\n**",ret);  }  return 0;  } |

.  
[14.hdu1542 Atlantis](http://acm.hdu.edu.cn/showproblem.php?pid=1542)  
矩形面积并,扫描线法(发现我们HDU的队员的矩形面积交模板基本都是在最坏情况下更新到底,宁波惨痛的教训啊..)

[?](http://www.ericbess.com/ericblog/2008/03/03/wp-codebox/#examples)[[Copy to clipboard]View Code](javascript:;) CPP

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107 | struct Seg\_Tree{  int left,right;  int cnt;  double sum;  int mid() {  return (left + right) >> 1;  }  }tt[6666];  struct Seg{  double high;  double left,right;  int side;  }ss[2222];  bool cmp(Seg &a, Seg &b) {  return a.high < b.high;  }  double pos[2222];    void build(int l,int r,int idx) {  tt[idx].left = l;  tt[idx].right = r;  tt[idx].cnt = 0;  tt[idx].sum = 0;  if(l == r) return ;  int mid = tt[idx].mid();  build(l,mid,LL(idx));  build(mid+1,r,RR(idx));  }    void update(int idx) {  if(tt[idx].cnt) {  tt[idx].sum = pos[tt[idx].right+1] - pos[tt[idx].left];  } else if(tt[idx].left == tt[idx].right) {  tt[idx].sum = 0;  } else {  tt[idx].sum = tt[LL(idx)].sum + tt[RR(idx)].sum;  }  }    void update(int l,int r,int st,int idx) {  if(l <= tt[idx].left && r >= tt[idx].right) {  tt[idx].cnt += st;  update(idx);  return ;  }  int mid = tt[idx].mid();  if(l <= mid) update(l,r,st,LL(idx));  if(mid < r) update(l,r,st,RR(idx));  update(idx);  }    int Bin(double x,int hi) {  int lo = 0;  while(lo <= hi) {  int mid = (lo + hi) >> 1;  if(pos[mid] == x) return mid;  if(pos[mid] < x) {  lo = mid + 1;  } else {  hi = mid - 1;  }  }  return -1;  }    int main() {  int n;  int cas = 1;  while(scanf("%d",&n) == 1) {  if(n == 0) break;  int m = 0;  FF(i,n) {  double a,b,c,d;  scanf("%lf%lf%lf%lf",&a,&b,&c,&d);  pos[m] = ss[m].left = ss[m+1].left = a;  ss[m].high = b;  pos[m+1] = ss[m].right = ss[m+1].right = c;  ss[m+1].high = d;  ss[m].side = 1;  ss[m+1].side = -1;  m += 2;  }  sort(ss,ss+m,cmp);  sort(pos,pos+m);  int k = 0;  FF(i,m) {  if(i == 0 || pos[i] != pos[i-1]) {  pos[k++] = pos[i];  }  }  build(0,k-1,1);  double ret = 0;  FF(i,m-1) {  int l = Bin(ss[i].left,k-1);  int r = Bin(ss[i].right,k-1) - 1;  if(l > r) {  ret += tt[1].sum \* (ss[i+1].high - ss[i].high);  continue;  }  update(l,r,ss[i].side,1);  ret += tt[1].sum \* (ss[i+1].high - ss[i].high);  }  printf("Test case #%d**\n**",cas++);  printf("Total explored area: %.2lf**\n\n**",ret);  }  return 0;  } |

.  
[15.hdu1255 覆盖的面积](http://acm.hdu.edu.cn/showproblem.php?pid=1255)  
同上,扫描线法,我多加了一个系数csum,来统计覆盖两次的长度(156MS,第一次做线段树排到第一,纪念下)

[?](http://www.ericbess.com/ericblog/2008/03/03/wp-codebox/#examples)[[Copy to clipboard]View Code](javascript:;) CPP

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121 | struct Seg\_Tree{  int left,right;  double sum;  double csum;  int cnt;  int mid() {  return (left + right) >> 1;  }  }tt[6000];  struct Seg{  double left,right,high;  int side;  }ss[2001];  int k;  double pos[2001];  bool cmp(Seg &a,Seg &b) {  return a.high < b.high;  }    void build(int l,int r,int idx) {  tt[idx].left = l;  tt[idx].right = r;  tt[idx].sum = 0;  tt[idx].csum = 0;  tt[idx].cnt = 0;  if(l == r) return ;  int mid = tt[idx].mid();  build(l,mid,LL(idx));  build(mid+1,r,RR(idx));  }    void update(int idx) {  if(tt[idx].cnt > 0) {  tt[idx].sum = pos[tt[idx].right + 1] - pos[tt[idx].left];  } else if(tt[idx].left == tt[idx].right) {  tt[idx].sum = 0;  } else {  tt[idx].sum = tt[LL(idx)].sum + tt[RR(idx)].sum;  }    if(tt[idx].cnt > 1) {  tt[idx].csum = pos[tt[idx].right + 1] - pos[tt[idx].left];  } else if(tt[idx].left == tt[idx].right) {  tt[idx].csum = 0;  } else if(tt[idx].cnt > 0) {  tt[idx].csum = tt[LL(idx)].sum + tt[RR(idx)].sum;  } else {  tt[idx].csum = tt[LL(idx)].csum + tt[RR(idx)].csum;  }  }    void update(int l,int r,int st,int idx) {  if(l <= tt[idx].left && r >= tt[idx].right) {  tt[idx].cnt += st;  update(idx);  return ;  }  int mid = tt[idx].mid();  if(l <= mid) update(l,r,st,LL(idx));  if(mid < r) update(l,r,st,RR(idx));  update(idx);  }    int Bin(double x) {  int hi = k - 1;  int lo = 0;  while(lo <= hi) {  int mid = (lo + hi) >> 1;  if(pos[mid] == x) {  return mid;  }  if(pos[mid] < x) {  lo = mid + 1;  } else {  hi = mid - 1;  }  }  return -1;  }  int main() {  int T;  scanf("%d",&T);  while(T --) {  int n;  scanf("%d",&n);  int m = 0;  FF(i,n) {  double a,b,c,d;  scanf("%lf%lf%lf%lf",&a,&b,&c,&d);  pos[m] = ss[m].left = ss[m+1].left = a;  ss[m].high = b;  pos[m+1] = ss[m].right = ss[m+1].right = c;  ss[m+1].high = d;  ss[m].side = 1;  ss[m+1].side = -1;  m += 2;  }  sort(ss,ss+m,cmp);  sort(pos,pos+m);  k = 0;  FF(i,m) {  if(i == 0 || pos[i] != pos[i-1]) {  pos[k++] = pos[i];  }  }  build(0,k-1,1);  double ans = 0;  FF(i,m-1) {  if(ss[i].left == ss[i].right) {  ans += tt[1].sum \* (ss[i+1].high - ss[i].high);  continue;  }  int l = Bin(ss[i].left);  int r = Bin(ss[i].right) - 1;  update(l,r,ss[i].side,1);  ans += tt[1].csum \* (ss[i+1].high - ss[i].high);  }  printf("%.2lf**\n**",ans);  }  return 0;  } |

.  
[16.hdu1828 Picture](http://acm.hdu.edu.cn/showproblem.php?pid=1828)  
扫描线,同面积统计,加了一个num\_Seg统计一个扫描区域里的边

[?](http://www.ericbess.com/ericblog/2008/03/03/wp-codebox/#examples)[[Copy to clipboard]View Code](javascript:;) CPP

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99 | struct Seg\_Tree{  int left,right;  int lbd,rbd;  int cnt;  int Len;  int num\_Seg;  int mid() {  return (left + right) >> 1;  }  }tt[66666];  struct Seg{  int left;  int right;  int high;  int side;  }ss[10001];    bool cmp(Seg &a,Seg &b) {  return a.high < b.high;  }    void build(int l,int r,int idx) {  tt[idx].left = l;  tt[idx].right = r;  tt[idx].cnt = tt[idx].Len = tt[idx].num\_Seg = 0;  if(l == r) return ;  int mid = tt[idx].mid();  build(l,mid,LL(idx));  build(mid+1,r,RR(idx));  }    void update(int idx) {  if(tt[idx].cnt) {  tt[idx].lbd = tt[idx].rbd = 1;  tt[idx].Len = tt[idx].right - tt[idx].left + 1;  tt[idx].num\_Seg = 2;  } else if(tt[idx].left == tt[idx].right) {  tt[idx].lbd = tt[idx].rbd = 0;  tt[idx].Len = 0;  tt[idx].num\_Seg = 0;  } else {  int l = LL(idx);  int r = RR(idx);  tt[idx].lbd = tt[l].lbd;  tt[idx].rbd = tt[r].rbd;  tt[idx].Len = tt[l].Len + tt[r].Len;  tt[idx].num\_Seg = tt[l].num\_Seg + tt[r].num\_Seg - 2 \* (tt[l].rbd & tt[r].lbd);  }  }    void update(int l,int r,int st,int idx) {  if(tt[idx].left >= l && tt[idx].right <= r) {  tt[idx].cnt += st;  update(idx);  return ;  }  int mid = tt[idx].mid();  if(l <= mid) update(l,r,st,LL(idx));  if(mid < r) update(l,r,st,RR(idx));  update(idx);  }    int main() {  int n;  while(scanf("%d",&n) == 1) {  if(n == 0) {  puts("0");  continue;  }  int lup = 10000;  int rup = -10000;  int m = 0;  FF(i,n) {  scanf("%d%d%d%d",&ss[m].left,&ss[m].high,&ss[m].right,&ss[m+1].high);  ss[m+1].left = ss[m].left;  ss[m+1].right = ss[m].right;;  ss[m].side = 1;  ss[m+1].side = -1;  lup = min(lup,ss[m].left);  rup = max(rup,ss[m].right);  m += 2;  }  sort(ss,ss+m,cmp);  build(lup,rup-1,1);  int ans = 0;  int buf = 0;  ss[m] = ss[m-1];  FF(i,m) {  if(ss[i].left != ss[i].right) {  update(ss[i].left,ss[i].right-1,ss[i].side,1);  }  ans += tt[1].num\_Seg \* (ss[i+1].high - ss[i].high);  ans += abs(tt[1].Len - buf);  buf = tt[1].Len;  }  printf("%d**\n**",ans);  }  return 0;  } |

.  
[17.pku1436 Horizontally Visible Segments](http://acm.pku.edu.cn/JudgeOnline/problem?id=1436)  
成段更新,成段询问(染色问题,坐标\*2后很简单,最后统计用暴力- -)

[?](http://www.ericbess.com/ericblog/2008/03/03/wp-codebox/#examples)[[Copy to clipboard]View Code](javascript:;) CPP

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104 | struct Seg\_Tree{  int left,right;  int col;  int mid() {  return (left + right) >> 1;  }  }tt[80000];  #define mix -1  struct Seg{  int pos,left,right;  }hh[8888];  vector <int> edge[8888];  int hash[8888];  int cnt[8888];  bool cmp(Seg &a,Seg &b) {  return a.pos < b.pos;  }    void build(int l,int r,int idx) {  tt[idx].left = l;  tt[idx].right = r;  tt[idx].col = mix;  if(l == r) return ;  int mid = tt[idx].mid();  build(l,mid,LL(idx));  build(mid+1,r,RR(idx));  }    void update(int l,int r,int id,int idx) {  if(l <= tt[idx].left && r >= tt[idx].right) {  tt[idx].col = id;  return ;  }  if(tt[idx].col != mix) {  tt[LL(idx)].col = tt[RR(idx)].col = tt[idx].col;  tt[idx].col = mix;  }  int mid = tt[idx].mid();  if(l <= mid) update(l,r,id,LL(idx));  if(mid < r) update(l,r,id,RR(idx));  }    void query(int l,int r,int id,int idx) {  if(tt[idx].col != mix) {  if(hash[tt[idx].col] != id) {  edge[tt[idx].col].push\_back(id);  hash[tt[idx].col] = id;  }  return ;  }  if(tt[idx].left == tt[idx].right) return ;  if(tt[idx].col != mix) {  tt[LL(idx)].col = tt[RR(idx)].col = tt[idx].col;  tt[idx].col = mix;  }  int mid = tt[idx].mid();  if(r <= mid) {  query(l,r,id,LL(idx));  } else if(mid < l) {  query(l,r,id,RR(idx));  } else {  query(l,mid,id,LL(idx));  query(mid+1,r,id,RR(idx));  }  }    int main() {  int T;  scanf("%d",&T);  while(T --) {  int n;  build(0,16000,1);  scanf("%d",&n);  FF(i,n) {  scanf("%d%d%d",&hh[i].left,&hh[i].right,&hh[i].pos);  hh[i].left \*= 2;  hh[i].right \*= 2;  edge[i].clear();  }  sort(hh,hh+n,cmp);  CC(hash,-1);  FF(i,n) {  query(hh[i].left,hh[i].right,i,1);  update(hh[i].left,hh[i].right,i,1);  }  int ans = 0;  FF(i,n) {  int m = sz(edge[i]);  FF(j,m) {  int idx = edge[i][j];  int mm = sz(edge[idx]);  FF(k,m) {  FF(l,mm) {  if(edge[i][k] == edge[idx][l]) {  ans ++;  }  }  }  }  }  printf("%d**\n**",ans);  }  return 0;  } |

.  
[18.pku3225 Help with Intervals](http://acm.pku.edu.cn/JudgeOnline/problem?id=3225)  
成段更新,总询问区间(有个异或操作比较新颖)

[?](http://www.ericbess.com/ericblog/2008/03/03/wp-codebox/#examples)[[Copy to clipboard]View Code](javascript:;) CPP

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122  123  124  125  126  127  128  129  130  131  132  133  134  135  136  137  138  139  140  141  142  143  144  145  146  147  148  149  150  151  152  153  154  155  156  157  158  159  160  161  162  163  164  165  166  167  168  169  170  171  172  173  174  175  176  177  178  179  180  181  182  183  184  185  186  187  188  189  190  191  192  193  194  195  196  197  198  199  200  201  202  203  204  205  206  207  208  209  210  211  212  213  214  215  216  217  218 | #define mix -1  struct Seg\_Tree{  int left;  int right;  int cover;  int change;  int mid() {  return (left + right) >> 1;  }  void Change() {  if(change) {  change = 0;  } else if(cover != mix) {  cover ^= 1;  } else {  change = 1;  }  }  }tt[800000];  bool hash[131072];    void build(int l,int r,int idx) {  tt[idx].left = l;  tt[idx].right = r;  tt[idx].cover = 0;  tt[idx].change = 0;  if(l == r) return ;  int mid = tt[idx].mid();  build(l,mid,LL(idx));  build(mid+1,r,RR(idx));  }    void Scanf(int &a,int &b,char str[]) {  int pos = 1;  while(isdigit(str[pos])) {  a = a \* 10 + str[pos] - '0';  pos ++;  }  pos ++;  while(isdigit(str[pos])) {  b = b \* 10 + str[pos] - '0';  pos ++;  }  if(str[0] == '(') {  a = a \* 2 + 1;  } else {  a = a \* 2;  }  if(str[pos] == ')') {  b = b \* 2 - 1;  } else {  b = b \* 2;  }  }    void Printf(int a,int b) {  if(a%2 == 0) {  printf("[%d,",a/2);  } else {  printf("(%d,",a/2);  }  if(b%2 == 0) {  printf("%d]",b/2);  } else {  printf("%d)",b/2+1);  }  }    void update(int l,int r,char op,int idx) {    int mid;  switch(op) {  case 'U':  if(l == tt[idx].left && r == tt[idx].right) {  tt[idx].cover = 1;  tt[idx].change = false;  return ;  }  if(tt[idx].cover != mix) {  tt[LL(idx)].cover = tt[RR(idx)].cover = tt[idx].cover;  tt[LL(idx)].change = tt[RR(idx)].change = 0;  tt[idx].cover = mix;  }  break;  case 'I':  if(l == tt[idx].left && r == tt[idx].right) {  return ;  }  if(tt[idx].cover != mix) {  tt[LL(idx)].cover = tt[RR(idx)].cover = tt[idx].cover;  tt[LL(idx)].change = tt[RR(idx)].change = 0;  tt[idx].cover = mix;  }  mid = tt[idx].mid();  if(r <= mid) tt[RR(idx)].cover = tt[idx].change;  if(mid < l) tt[LL(idx)].cover = tt[idx].change;  break;  case 'D':  if(l == tt[idx].left && r == tt[idx].right) {  tt[idx].cover = 0;  tt[idx].change = false;  return ;  }  if(tt[idx].cover != mix) {  tt[LL(idx)].cover = tt[RR(idx)].cover = tt[idx].cover;  tt[LL(idx)].change = tt[RR(idx)].change = 0;  tt[idx].cover = mix;  }  break;  case 'C':  if(l == tt[idx].left && r == tt[idx].right) {  tt[idx].Change();  return ;  }  if(tt[idx].cover != mix) {  tt[LL(idx)].cover = tt[RR(idx)].cover = tt[idx].cover;  tt[LL(idx)].change = tt[RR(idx)].change = 0;  tt[idx].cover = mix;  }  mid = tt[idx].mid();  if(r <= mid) tt[RR(idx)].cover = tt[idx].change;  if(mid < l) tt[LL(idx)].cover = tt[idx].change;  break;  case 'S':  if(l == tt[idx].left && r == tt[idx].right) {  tt[idx].Change();  return ;  }  if(tt[idx].cover != mix) {  tt[LL(idx)].cover = tt[RR(idx)].cover = tt[idx].cover;  tt[LL(idx)].change = tt[RR(idx)].change = 0;  tt[idx].cover = mix;  }  break;  }  if(tt[idx].change) {  tt[LL(idx)].Change();  tt[RR(idx)].Change();  tt[idx].change = 0;  }  mid = tt[idx].mid();  if(r <= mid) {  update(l,r,op,LL(idx));  } else if(mid < l) {  update(l,r,op,RR(idx));  } else {  update(l,mid,op,LL(idx));  update(mid+1,r,op,RR(idx));  }  if(tt[LL(idx)].cover == tt[RR(idx)].cover) {  tt[idx].cover = tt[LL(idx)].cover;  }  }    void query(int l,int r,int idx) {  if(l == r) {  if(tt[idx].cover) {  hash[l] = true;  }  return ;  }  if(tt[idx].cover != mix) {  tt[LL(idx)].cover = tt[RR(idx)].cover = tt[idx].cover;  tt[LL(idx)].change = tt[RR(idx)].change = 0;  tt[idx].cover = mix;  }  if(tt[idx].change) {  tt[LL(idx)].Change();  tt[RR(idx)].Change();  tt[idx].change = 0;  }  int mid = tt[idx].mid();  query(l,mid,LL(idx));  query(mid+1,r,RR(idx));  }    int M = 131072;  int main() {  build(0,M,1);  char op[2];  char str[99];  while(scanf("%s%s",op,str) == 2) {  int a(0),b(0);  Scanf(a,b,str);  if(a > b) {  if(op[0] == 'C' || op[0] == 'I') {  tt[1].cover = 0;  tt[1].change = 0;  }  continue;  }  update(a,b,op[0],1);  }  CC(hash,false);  query(0,M,1);  bool flag = false;  int start = -1,end;  FF(i,M) {  if(hash[i]) {  if(start == -1) {  start = i;  }  end = i;  } else {  if(start != -1) {  if(flag) printf(" ");  flag = true;  Printf(start,end);  start = -1;  }  }  }  if(!flag) {  printf("empty set");  }  puts("");  return 0;  } |

.  
[19.pku2482 Stars in Your Window](http://acm.pku.edu.cn/JudgeOnline/problem?id=2482)  
成段更新,区间最值 + 扫描线(转化成区间最值有点巧妙,数据太TMD的恶心了,中间二分的地方会int溢出,检查了半个小时)

[?](http://www.ericbess.com/ericblog/2008/03/03/wp-codebox/#examples)[[Copy to clipboard]View Code](javascript:;) CPP

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106 | struct Seg\_Tree{  int left,right;  int val;  int add;  int mid() {  return (left + right)>>1;  }  }tt[40000];  struct star{  int x,y,val;  }hh[10001];  int pos[10001];  int n , m , H, W;  bool cmp(star &a,star &b) {  return a.y < b.y;  }    int Bin(LL x) {  int lo = 0;  int hi = m - 1;  while(lo <= hi) {  int mid = (lo + hi) >> 1;  if(pos[mid] < x) {  lo = mid + 1;  } else {  hi = mid - 1;  }  }  return lo;  }    void build(int l,int r,int idx) {  tt[idx].left = l;  tt[idx].right = r;  tt[idx].add = 0;  tt[idx].val = 0;  if(l == r) return ;  int mid = tt[idx].mid();  build(l,mid,LL(idx));  build(mid+1,r,RR(idx));  }    void update(int l,int r,int add,int idx) {  if(l <= tt[idx].left && r >= tt[idx].right) {  tt[idx].add += add;  tt[idx].val += add;  return ;  }  if(tt[idx].add) {  tt[LL(idx)].add += tt[idx].add;  tt[RR(idx)].add += tt[idx].add;  tt[LL(idx)].val += tt[idx].add;  tt[RR(idx)].val += tt[idx].add;  tt[idx].add = 0;  }  int mid = tt[idx].mid();  if(l <= mid) update(l,r,add,LL(idx));  if(mid < r) update(l,r,add,RR(idx));  tt[idx].val = max(tt[LL(idx)].val , tt[RR(idx)].val);  }    int main() {  while(scanf("%d%d%d",&n,&W,&H) == 3) {  FF(i,n) {  scanf("%d%d%d",&hh[i].x,&hh[i].y,&hh[i].val);  pos[i] = hh[i].x;  }  sort(hh,hh+n,cmp);  sort(pos,pos+n);  m = 0;  FF(i,n) {  if(i == 0 || pos[i] != pos[i-1]) {  pos[m++] = pos[i];  }  }  build(0,m-1,1);  int start = 0,end = 0;  int ans = 0;  FF(i,n) {  if(i && hh[i].y == hh[i-1].y) continue;  FOR(j,start,i) {  update(Bin(hh[j].x) , Bin(hh[j].x + W) - 1, -hh[j].val , 1);  }  start = i;  int lo = i;  int hi = n-1;  LL buf = (LL)hh[i].y + H;  while(lo <= hi) {  int mid = (lo + hi) >> 1;  if(hh[mid].y < buf) {  lo = mid + 1;  } else {  hi = mid - 1;  }  }  FOR(j,end,lo) {  update(Bin(hh[j].x) , Bin((LL)hh[j].x + W) - 1, hh[j].val , 1);  }  end = lo;  checkmax(ans,tt[1].val);  if(end == n) break;  }  printf("%d**\n**",ans);  }  return 0;  } |

.  
[20.pku2828 Buy Tickets](http://acm.pku.edu.cn/JudgeOnline/problem?id=2828)  
思维很巧妙,倒过来做的话就能确定此人所在的位置

[?](http://www.ericbess.com/ericblog/2008/03/03/wp-codebox/#examples)[[Copy to clipboard]View Code](javascript:;) CPP

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51 | struct Seg\_Tree{  int left,right;  int remain;  int mid() {  return (left + right) >> 1;  }  }tt[600000];    int pos[200001];  int newpos[200001];  int val[200001];    void build(int l,int r,int idx) {  tt[idx].left = l;  tt[idx].right = r;  tt[idx].remain = r - l + 1;  if(l == r) return ;  int mid = tt[idx].mid();  build(l,mid,LL(idx));  build(mid+1,r,RR(idx));  }    int query(int id,int idx) {  tt[idx].remain --;  if(tt[idx].left == tt[idx].right) {  return tt[idx].left;  }  if(tt[LL(idx)].remain >= id) {  return query(id,LL(idx));  } else {  return query(id-tt[LL(idx)].remain,RR(idx));  }  }    int main() {  int n;  while(scanf("%d",&n) == 1) {  build(0,n-1,1);  FF(i,n) {  scanf("%d%d",&pos[i],&val[i]);  }  FFD(i,n) {  newpos[query(pos[i]+1,1)] = i;  }  FF(i,n-1) {  printf("%d ",val[newpos[i]]);  }  printf("%d**\n**",val[newpos[n-1]]);  }  return 0;  } |

.  
[21.pku2464 Brownie Points II](http://acm.pku.edu.cn/JudgeOnline/problem?id=2464)  
更新节点,区间求和 + 扫描线(很好玩的一道题目,用两个线段树沿着扫描线更新,然后按”自己最多,对方最少”的方案一路统计)  
(因为两棵树,我就直接写成类了)

[?](http://www.ericbess.com/ericblog/2008/03/03/wp-codebox/#examples)[[Copy to clipboard]View Code](javascript:;) CPP

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122  123  124  125  126  127  128  129  130  131  132  133  134  135  136  137  138  139  140  141 | class Key {  private:  struct Seg\_Tree{  int left;  int right;  int sum;  int mid() {  return (left + right) >> 1;  }  }tt[600000];  public:  void build(int l,int r,int idx) {  tt[idx].left = l;  tt[idx].right = r;  tt[idx].sum = 0;  if(l == r) return ;  int mid = tt[idx].mid();  build(l,mid,LL(idx));  build(mid+1,r,RR(idx));  }  void update(int pos,int st,int idx) {  if(tt[idx].left == tt[idx].right) {  tt[idx].sum += st;  return ;  }  int mid = tt[idx].mid();  if(pos <= mid) {  update(pos,st,LL(idx));  } else {  update(pos,st,RR(idx));  }  tt[idx].sum = tt[LL(idx)].sum + tt[RR(idx)].sum;  }  int query(int l,int r,int idx) {  if(l == tt[idx].left && r == tt[idx].right) {  return tt[idx].sum;  }  int mid = tt[idx].mid();  if(r <= mid) {  return query(l,r,LL(idx));  } else if(mid < l) {  return query(l,r,RR(idx));  } else {  return query(l,mid,LL(idx)) + query(mid+1,r,RR(idx));  }  }  }Left,Right;    struct Point {  int x,y;  }p[200001];  int pos[200010];  bool cmp(Point &a,Point &b) {  return a.x < b.x;  }  int m;  int Bin(int x) {  int lo = 1;  int hi = m - 1;  while(lo <= hi) {  int mid = (lo + hi) >> 1;  if(pos[mid] == x) {  return mid;  }  if(pos[mid] < x) {  lo = mid + 1;  } else {  hi = mid - 1;  }  }  return -1;  }    vector <int> ans;    int main() {  int n;  while(scanf("%d",&n) == 1) {  if(n == 0) break;  FF(i,n) {  scanf("%d%d",&p[i].x,&p[i].y);  pos[i+1] = p[i].y;  }  sort(pos+1,pos+n+1);  sort(p,p+n,cmp);  m = 1;  FOR(i,1,n+1) {  if(i == 1 || pos[i] != pos[i-1]) {  pos[m++] = pos[i];  }  }  Left.build(0,m,1);  Right.build(0,m,1);  FF(i,n) {  Right.update(Bin(p[i].y),1,1);  }  int start = 0;  int best = 0;  ans.clear();  FOR(i,1,n+1) {  if(i == n || p[i].x != p[i-1].x) {  FOR(j,start,i) {  Right.update(Bin(p[j].y),-1,1);  }  int Ollie = -1;  int Stan = -1;  FOR(j,start,i) {  int idx = Bin(p[j].y);  int A = Left.query(0,idx-1,1) + Right.query(idx+1,m,1);  int B = Left.query(idx+1,m,1) + Right.query(0,idx-1,1);  if(Ollie == B) {  Stan = min(Stan , A);  } else if(Ollie < B) {  Ollie = B;  Stan = A;  }  }  if(Stan > best) {  best = Stan;  ans.clear();  }  if(Stan == best) {  ans.push\_back(Ollie);  }  FOR(j,start,i) {  Left.update(Bin(p[j].y),1,1);  }  start = i;  }  }  printf("Stan: %d; Ollie:",best);  sort(ans.begin(),ans.end());  FF(i,sz(ans)) {  if(i == 0 || ans[i] != ans[i-1]) {  printf(" %d",ans[i]);  }  }  puts(";");  }  return 0;  } |

.  
[22.pku3145 Harmony Forever](http://acm.pku.edu.cn/JudgeOnline/problem?id=3145)  
查找一个区间内最左边的数,询问的val大的话用线段树,小的话线性扫描,很囧的题目

[?](http://www.ericbess.com/ericblog/2008/03/03/wp-codebox/#examples)[[Copy to clipboard]View Code](javascript:;) CPP

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122  123  124  125  126  127  128  129  130 | struct Seg\_Tree{  int left,right;  bool exist;  bool unexist;  int mid() {  return (left + right) >> 1;  }  }tt[2000000];  int num[40010];  int rank[5000001];  #define M 50000  void build(int l,int r,int idx) {  tt[idx].left = l;  tt[idx].right = r;  tt[idx].exist = false;  tt[idx].unexist = false;  if(l == r) return ;  int mid = tt[idx].mid();  build(l,mid,LL(idx));  build(mid+1,r,RR(idx));  }    void update(int id,int idx) {  tt[idx].exist = true;  if(tt[idx].left == tt[idx].right) {  return ;  }  if(tt[idx].unexist) {  tt[LL(idx)].unexist = tt[RR(idx)].unexist = true;  tt[LL(idx)].exist = tt[RR(idx)].exist = false;  tt[idx].unexist = false;  }  int mid = tt[idx].mid();  if(id <= mid) {  update(id,LL(idx));  } else {  update(id,RR(idx));  }  }    int query(int l,int r,int idx) {  if(tt[idx].left == tt[idx].right) {  return tt[idx].left;  }  if(tt[idx].unexist) {  tt[LL(idx)].unexist = tt[RR(idx)].unexist = true;  tt[LL(idx)].exist = tt[RR(idx)].exist = false;  tt[idx].unexist = false;  }  int mid = tt[idx].mid();  int buf = -1;  if(l <= mid) {  if(tt[LL(idx)].exist) {  buf = query(l,r,LL(idx));  }  }  if(buf != -1) return buf;  if(mid < r) {  if(tt[RR(idx)].exist) {  buf = query(l,r,RR(idx));  }  }  return buf;  }    int main() {  build(0,500000,1);  int T, cas = 1;  while(scanf("%d",&T) == 1) {  if(T == 0) break;  if(cas != 1) {  puts("");  }  printf("Case %d:**\n**",cas++);  tt[1].unexist = true;  int n = 0;  while(T --) {  char op[2];  int val;  scanf("%s%d",op,&val);  if(op[0] == 'B') {  update(val,1);  num[n++] = val;  rank[val] = n;  } else {  if(n == 0) {  puts("-1");  continue;  }  if(val <= 1000) {  int remain = val;  int last = -1;  FFD(i,n) {  int buf = num[i] % val;  if(buf < remain) {  remain = buf;  last = i+1;  }  if(remain == 0) {  break;  }  }  printf("%d**\n**",last);  } else {  int remain = val;  int last = -1;  int start = 0;  int end = val - 1;  while(start <= 500000) {  checkmin(end,500000);  int key = query(start,end,1);  if(key != -1) {  int buf = key % val;  if(buf < remain) {  remain = buf;  last = rank[key];  } else if(buf == remain) {  checkmax(last,rank[key]);  }  }  start += val;  end += val;  }  printf("%d**\n**",last);  }  }  }  }  return 0;  } |

.  
[23.pku2886 Who Gets the Most Candies?](http://acm.pku.edu.cn/JudgeOnline/problem?id=2886)  
寻找区间中的左数第N个数,约瑟夫环(学到了反素数表,可以不用把所有数字预处理出来了)

[?](http://www.ericbess.com/ericblog/2008/03/03/wp-codebox/#examples)[[Copy to clipboard]View Code](javascript:;) CPP

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63 | struct Seg\_Tree{  int left,right;  int sum;  int mid() {  return (left + right) >> 1;  }  }tt[1100000];  struct H{  char name[12];  int val;  }hh[500001];    void build(int l,int r,int idx) {  tt[idx].left = l;  tt[idx].right = r;  tt[idx].sum = (r - l + 1);  if(l == r) return ;  int mid = tt[idx].mid();  build(l,mid,LL(idx));  build(mid+1,r,RR(idx));  }    int update(int k,int idx) {  tt[idx].sum --;  if(tt[idx].left == tt[idx].right) {  return tt[idx].left;  }  int mid = tt[idx].mid();  if(k <= tt[LL(idx)].sum) return update(k,LL(idx));  return update(k-tt[LL(idx)].sum,RR(idx));  }    const int antiprime[] = {1,2,4,6,12,24,36,48,60,120,180,240,360,720,840,1260,1680,2520,5040,7560,10080,15120,20160,25200,27720,45360,50400,55440,83160,110880,166320, 221760, 277200, 332640, 498960, 554400, 665280};  const int factorNum[] = {1, 2, 3, 4, 6, 8, 9, 10, 12, 16, 18, 20, 24, 30, 32, 36, 40, 48, 60, 64, 72, 80, 84, 90, 96, 100, 108, 120, 128, 144, 160, 168, 180, 192, 200, 216, 224};    int main() {  int n , k, &mod = tt[1].sum;  while(scanf("%d%d",&n,&k) == 2) {  FOR(i,1,n+1) {  scanf("%s%d",hh[i].name,&hh[i].val);  }  build(1,n,1);    int tag=0;  while(antiprime[tag] <= n) {  tag++;  }  int need = antiprime[--tag];    int pos = 0;  hh[pos].val = 0;  while(need --) {  if(hh[pos].val > 0) {  k = ((k + hh[pos].val - 2)%mod + mod)%mod + 1;  } else {  k = ((k + hh[pos].val - 1)%mod + mod)%mod + 1;  }  pos = update(k,1);  }  printf("%s %d**\n**",hh[pos].name,factorNum[tag]);  }  return 0;  } |

.  
[24.pku2991 Crane](http://acm.pku.edu.cn/JudgeOnline/problem?id=2991)  
记录了线段的两端点以及转过的角度,成段的转,超有意思的题目,做了之后会对线段树理解更深刻  
(wy教主推荐的,不过我的代码写的太搓了..没啥学习的价值)

[?](http://www.ericbess.com/ericblog/2008/03/03/wp-codebox/#examples)[[Copy to clipboard]View Code](javascript:;) CPP

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122  123  124  125  126  127  128  129  130  131 | #define M 10001  struct Point {  double x,y;  Point operator + (Point a) {  Point ret;  ret.x = x + a.x;  ret.y = y + a.y;  return ret;  }  Point operator - (Point a) {  Point ret;  ret.x = x - a.x;  ret.y = y - a.y;  return ret;  }  };  struct Seg\_Tree{  int left,right;  int angle;  Point s,e;  bool cover;  int mid() {  return (left + right) >> 1;  }  void turn(int angle) {  Point ret = s;  Point buf = s;  e.x -= s.x;  e.y -= s.y;  buf.x = cos(angle\*Pi/180);  buf.y = sin(angle\*Pi/180);  ret.x += e.x\*buf.x - e.y\*buf.y;  ret.y += e.x\*buf.y + e.y\*buf.x;  e = ret;  }  }tt[M\*4];  int len[M];  int sum[M];  void build(int l,int r,int idx) {  tt[idx].left = l;  tt[idx].right = r;  tt[idx].s.x = 0;  tt[idx].s.y = sum[l]-len[l];  tt[idx].e.x = 0;  tt[idx].e.y = sum[r];  tt[idx].angle = 0;  tt[idx].cover = false;  if(l == r) return ;  int mid = tt[idx].mid();  build(l,mid,LL(idx));  build(mid+1,r,RR(idx));  }    void update(int idx) {  int l = LL(idx);  int r = RR(idx);  tt[l].angle = (tt[l].angle + tt[idx].angle)%360;  tt[l].e = tt[l].e + tt[idx].s - tt[l].s;  tt[l].s = tt[idx].s;  tt[l].turn(tt[idx].angle);  tt[l].cover = true;    tt[r].e = tt[r].e + tt[l].e - tt[r].s;  tt[r].s = tt[l].e;  tt[r].angle = (tt[r].angle + tt[idx].angle)%360;  tt[r].turn(tt[idx].angle);  tt[r].cover = true;    tt[idx].angle = 0;  tt[idx].cover = false;  }    void update(int l,int a,int idx) {  if(tt[idx].left == l) {  tt[idx].angle = (tt[idx].angle + a)%360;  tt[idx].turn(a);  tt[idx].cover = true;  return ;  }  if(tt[idx].cover) {  update(idx);  }  int mid = tt[idx].mid();  if(l <= mid) {  update(l,a,LL(idx));  tt[RR(idx)].e = tt[RR(idx)].e + tt[LL(idx)].e - tt[RR(idx)].s;  tt[RR(idx)].s = tt[LL(idx)].e;  update(mid+1,a,RR(idx));  } else {  update(l,a,RR(idx));  }  tt[idx].e = tt[RR(idx)].e;  }    int query(int id,int idx) {  if(tt[idx].left == tt[idx].right) {  return tt[idx].angle;  }  if(tt[idx].cover) {  update(idx);  }  int mid = tt[idx].mid();  if(id <= mid) {  return query(id,LL(idx));  } else {  return query(id,RR(idx));  }  tt[idx].e = tt[RR(idx)].e;  }    int main() {  int n , m;  int cas = 0;  while(scanf("%d%d",&n,&m) == 2) {  if(cas) puts("");  cas ++;  FOR(i,1,n+1) {  scanf("%d",&len[i]);  sum[i] = sum[i-1] + len[i];  }  build(1,n,1);  while(m --) {  int id,a;  scanf("%d%d",&id,&a);  a = ((180 - query(id+1,1) + query(id,1) + a) % 360 + 360)%360;  update(id+1,a,1);  printf("%.2lf %.2lf**\n**",tt[1].e.x+eps,tt[1].e.y+eps);  }  }  return 0;  } |

.  
[25.hdu1823 Luck and Love](http://acm.hdu.edu.cn/showproblem.php?pid=1823)  
二维线段树,没啥意思,代码量大了一倍而已,题目和运用范围都没一维的广,随便找了道题目练习下就好

[?](http://www.ericbess.com/ericblog/2008/03/03/wp-codebox/#examples)[[Copy to clipboard]View Code](javascript:;) CPP

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112 | struct Sub\_Seg\_Tree {  int left;  int right;  int val;  int mid() {  return (left + right) >> 1;  }  };  struct Seg\_Tree{  int left;  int right;  Sub\_Seg\_Tree tt[1001\*4];  int mid() {  return (left + right) >> 1;  }  }tt[101\*4];    void build2(Sub\_Seg\_Tree tt[],int l,int r,int idx) {  tt[idx].left = l;  tt[idx].right = r;  tt[idx].val = -1;  if(l == r) return ;  int mid = tt[idx].mid();  build2(tt,l,mid,LL(idx));  build2(tt,mid+1,r,RR(idx));  }    void build(int l,int r,int idx) {  build2(tt[idx].tt,0,1000,1);  tt[idx].left = l;  tt[idx].right = r;  if(l == r) return ;  int mid = tt[idx].mid();  build(l,mid,LL(idx));  build(mid+1,r,RR(idx));  }    void update2(Sub\_Seg\_Tree tt[],int b,int val,int idx) {  if(tt[idx].left == tt[idx].right) {  checkmax(tt[idx].val,val);  return ;  }  if(b <= tt[idx].mid()) {  update2(tt,b,val,LL(idx));  } else {  update2(tt,b,val,RR(idx));  }  tt[idx].val = max(tt[LL(idx)].val,tt[RR(idx)].val);  }    void update(int a,int b,int val,int idx) {  update2(tt[idx].tt,b,val,1);  if(tt[idx].left == tt[idx].right) return ;  if(a <= tt[idx].mid()) {  update(a,b,val,LL(idx));  } else {  update(a,b,val,RR(idx));  }  }    int query2(Sub\_Seg\_Tree tt[],int l,int r,int idx) {  if(l <= tt[idx].left && r >= tt[idx].right) {  return tt[idx].val;  }  int mid = tt[idx].mid();  int ret = -1;  if(l <= mid) checkmax(ret,query2(tt,l,r,LL(idx)));  if(mid < r) checkmax(ret,query2(tt,l,r,RR(idx)));  return ret;  }    int query(int l,int r,int l2,int r2,int idx) {  if(l <= tt[idx].left && r >= tt[idx].right) {  return query2(tt[idx].tt,l2,r2,1);  }  int mid = tt[idx].mid();  int ret = -1;  if(l <= mid) checkmax(ret,query(l,r,l2,r2,LL(idx)));  if(mid < r) checkmax(ret,query(l,r,l2,r2,RR(idx)));  return ret;  }    int main() {  int T;  while(scanf("%d",&T) == 1) {  if(T == 0) break;  build(100,200,1);  while(T --) {  char op;  while(!isalpha(op = getchar()));  if(op == 'I') {  int a;  double b,c;  scanf("%d%lf%lf",&a,&b,&c);  update(a,int(b\*10),int(c\*10),1);  } else {  int l1,r1;  double l2,r2;  scanf("%d%d%lf%lf",&l1,&r1,&l2,&r2);  if(l1 > r1) swap(l1,r1);  if(l2 > r2) swap(l2,r2);  int ret = query(l1,r1,int(l2\*10),int(r2\*10),1);  if(ret < 0) {  puts("-1");  } else {  printf("%.1lf**\n**",ret/10.0);  }  }  }  }  return 0;  } |

Popularity: 100% [[?](http://alexking.org/projects/wordpress/popularity-contest)]