segmenttree.cpp (4) dynamicallocation

Problem
Supports the operation of incrementing a paragraph into x units and finding the max of a segment. Use the launch level.
Complexity
per query: O (logn)
memory: min (2 * n, mlogn)
(n is the number of elements to be managed, m is the number of queries)

This code by Nguyen Tien Trung Kien

```
#include <stdio.h>
#include <iostream>
#include <algorithm>
using namespace std;

struct node {
   int ll, rr, max, Lazy;
   * * Left, * Right;

   node (int _ll, int _rr, int _Max) {ll = _ll, rr = _rr; Max =
_Max; }

   void diffuse () {
      if (ll! = rr) {
         if (Left) = new node (ll, (ll + rr) / 2, Max);
         if (Right) Right = new node ((ll + rr) / 2 + 1, rr, Max);
      }
      if (Lazy) {
         Max + = Lazy;
         if (Left) Left-> Lazy + = Lazy;
         if (Right) Right-> Lazy + = Lazy;
         Lazy = 0;
      }
   }

   void update (int L, int R, int X) {
      diffuse ();
      if (L> rr || R <ll) return;
      if (L <= ll && rr <= R) {Lazy + = X; diffuse (); }
      else {
         Left-> update (L, R, X);
         Right-> update (L, R, X);
         Max = max (Left-> Max, Right-> Max);
      }
   }

   int max_range (int L, int R) {
      diffuse ();
      if (L> rr || R <ll) return -0x3c3c3c3c;
      if (L <= ll && rr <= R) return Max;
      else return max (Left-> max_range (L, R), Right-> max_range (L,
R));
   }
};
```

```
int n, m;
node * root;

main () {
   ios :: sync_with_stdio (false);
   cin >> n >> m;
   root = new node (1, n, 0);
   while (m -) {
      int x, y, z, ch;
      cin >> ch >> x >> y;
      if (ch == 1) cout << root-> max_range (x, y) << endl;
      else {cin >> z; root-> update (x, y, z); }

   }
}
```

Comment

The tree segment uses a dynamic range that can be used to solve for very large n items (n <= 10 ^ 9).
This code was used to submit a post on SPOJ.

Refer

http://go.spoj.com/problems/QMAX2/