

Athugasemdir

Bergur Snorrason

17. febrúar 2022

- ▶ Á mánudaginn verður ekki fyrirlestur.
- ▶ Þess í stað verður miðmisseriskeppnin.
- ▶ Vikuskilin verða ekki heldur af hefðbundnu sniði.
- ▶ Keppnin hefst klukkan 8 : 30 og lýkur 13 : 30.
- ▶ Við ætlumst ekki til þess að þið séuð að allan tímann.
- ▶ Til að fá vikuskil í næstu viku nægir heiðarleg tilraun við eitt dæmi.
- ▶ Síðan má fá aukaskil með því að leysa fjögur dæmi fyrir sunnudaginn.
- ▶ Næsta miðvikudag munum við fara yfir lausnir á dæmunum í keppninni.

- ▶ Ég er búinn að breyta kóðanum fyrir biltréin örlítið.
- ▶ Það var mjög auðvelt að gera lúmska villu.

Gamla með villunni

```
8 int p[MAXN*5], n;
9 int qrec(int i, int j, int x, int y, int e) // Hjálparfall.
10 { // Við erum að leita að bili [x, y] og erum í [i, j].
11     if (x == i && y == j) return p[e];
12     int m = (i + j)/2;
13     if (y <= m) return qrec(i, m, x, y, LEFT(e));
14     if (x > m) return qrec(m + 1, j, x, y, RIGHT(e));
15     return qrec(i, m, x, m, LEFT(e)) + qrec(m + 1, j, m + 1, y, RIGHT(e));
16 }
17 int query(int x, int y) { return qrec(0, n - 1, x, y, 1); }
18 void urec(int i, int j, int x, int y, int e) // Hjálparfall.
19 { // Við erum að leita að laufinu [x, x] og erum í [i, j].
20     if (i == j) p[e] += y;
21     else
22     {
23         int m = (i + j)/2;
24         if (x <= m) urec(i, m, x, y, LEFT(e));
25         else urec(m + 1, j, x, y, RIGHT(e));
26         p[e] = p[LEFT(e)] + p[RIGHT(e)];
27     }
28 }
29 void update(int x, int y) { return urec(0, n - 1, x, y, 1); }
30 int main()
31 {
32     int i, x, y, z, n, q;
33     scanf("%d%d", &n, &q);
34     for (i = 0; i < 5*n; i++) p[i] = 0;
35     while (q-- != 0)
36     {
37         scanf("%d%d%d", &x, &y, &z);
38         x == 1 ? update(y, z) : printf("%d\n", query(y, z));
39     }
40     return 0;
41 }
```

Gamla án villunnar

```
8 int p[MAXN*5], n;
9 int qrec(int i, int j, int x, int y, int e) // Hjálparfall.
10 { // Við erum að leita að bili [x, y] og erum í [i, j].
11     if (x == i && y == j) return p[e];
12     int m = (i + j)/2;
13     if (y <= m) return qrec(i, m, x, y, LEFT(e));
14     if (x > m) return qrec(m + 1, j, x, y, RIGHT(e));
15     return qrec(i, m, x, m, LEFT(e)) + qrec(m + 1, j, m + 1, y, RIGHT(e));
16 }
17 int query(int x, int y) { return qrec(0, n - 1, x, y, 1); }
18 void urec(int i, int j, int x, int y, int e) // Hjálparfall.
19 { // Við erum að leita að laufinu [x, x] og erum í [i, j].
20     if (i == j) p[e] += y;
21     else
22     {
23         int m = (i + j)/2;
24         if (x <= m) urec(i, m, x, y, LEFT(e));
25         else urec(m + 1, j, x, y, RIGHT(e));
26         p[e] = p[LEFT(e)] + p[RIGHT(e)];
27     }
28 }
29 void update(int x, int y) { return urec(0, n - 1, x, y, 1); }
30 int main()
31 {
32     int i, x, y, z, q;
33     scanf("%d%d", &n, &q);
34     for (i = 0; i < 5*n; i++) p[i] = 0;
35     while (q-- != 0)
36     {
37         scanf("%d%d%d", &x, &y, &z);
38         x == 1 ? update(y, z) : printf("%d\n", query(y, z));
39     }
40     return 0;
41 }
```

Nýja

```
8 int p[MAXN*5];
9 int qrec(int i, int j, int x, int y, int e) // Hjálparfall.
10 { // Við erum að leita að bili [x, y] og erum í [i, j].
11     if (x == i && y == j) return p[e];
12     int m = (i + j)/2;
13     if (y <= m) return qrec(i, m, x, y, LEFT(e));
14     if (x > m) return qrec(m + 1, j, x, y, RIGHT(e));
15     return qrec(i, m, x, m, LEFT(e)) + qrec(m + 1, j, m + 1, y, RIGHT(e));
16 }
17 int query(int x, int y)
18 { // Finnum summuna yfir [x, y].
19     return qrec(0, p[0] - 1, x, y, 1);
20 }
21 void urec(int i, int j, int x, int y, int e) // Hjálparfall.
22 { // Við erum að leita að laufinu [x, x] og erum í [i, j].
23     if (i == j) p[e] += y;
24     else
25     {
26         int m = (i + j)/2;
27         if (x <= m) urec(i, m, x, y, LEFT(e));
28         else urec(m + 1, j, x, y, RIGHT(e));
29         p[e] = p[LEFT(e)] + p[RIGHT(e)];
30     }
31 }
32 void update(int x, int y)
33 { // Bætum y við x-ta stakið.
34     return urec(0, p[0] - 1, x, y, 1);
35 }
36
37 void init(int n)
38 {
39     for (int i = 0; i < 5*n; i++) p[i] = 0;
40     p[0] = n;
41 }
```

- ▶ Breytingin er að geyma stærðina á trénu í staki `p[0]`.
- ▶ Við erum einmitt að nota, svo kallaða, „fyrri aðferð“ til að geyma tréð í fylki.
- ▶ Þá er það stak ekki notað.

Nýja

```
43 int main()
44 {
45     int i, x, y, z, n, q;
46     scanf("%d%d", &n, &q);
47     init(n);
48     while (q-- != 0)
49     {
50         scanf("%d%d%d", &x, &y, &z);
51         x == 1 ? update(y, z) : printf("%d\n", query(y, z));
52     }
53     return 0;
54 }
```


