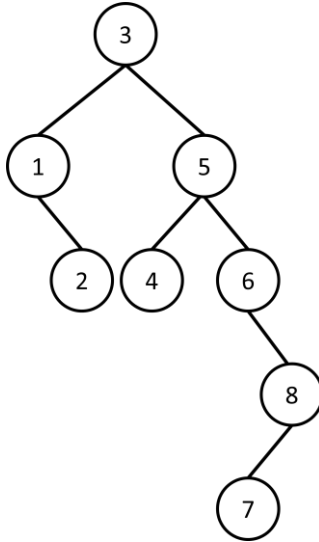


Binary Search Tree – Insert Counter (T0302)



Binary Search Tree merupakan pohon yang setiap node-nya memiliki maksimal dua node saja, dan setiap node memenuhi dua syarat:

- (1) semua node pada sub pohon kiri nilainya lebih kecil
- (2) setiap node pada sub pohon kanan nilainya lebih besar.

Operasi insertion pada BST telah dijelaskan pada slide (versi rekursif) dan pada modul (versi iteratif). Pada soal ini anda diminta untuk menambahkan sebuah counter global untuk menghitung berapa node yang telah dikunjungi oleh operasi insertion.

Sebagai contoh, jika angka 3, 5, 1, 6, 8, 7, 2, 4 dimasukkan ke dalam BST secara berturut-turut, maka pohon yang dihasilkan akan nampak seperti gambar di sebelah kiri. Ketika angka 3 dimasukkan, tree masih kosong, sehingga tidak ada node yang dikunjungi (counter bertambah 0). Ketika angka 5 dimasukkan, operasi insert mengunjungi node 3, baru kemudian memasukkan node 5 sebagai anak kanannya (counter bertambah 1). Ketika angka 6 dimasukkan, operasi insert mengunjungi node 3 dan 5 (counter bertambah 2). Demikian seterusnya.

Buatlah sebuah program yang menerima input deretan angka yang ingin dimasukkan ke BST dan menampilkan nilai counter setiap kali selesai memasukkan satu angka.

Spesifikasi Input

Input diawali dengan sebuah bilangan bulat N ($1 \leq N \leq 1.000$) yang menunjukkan jumlah angka yang ingin dimasukkan ke BST. N baris berikutnya masing-masing berisi sebuah bilangan bulat m_i yang nilainya unik ($1 \leq m_i \leq N$).

Spesifikasi Output

Output terdiri dari N buah baris, masing-masing berisi nilai counter setelah data ke- i dimasukkan ke BST. Nilai counter mula-mula adalah 0.

Contoh Input

8
3
5
1
6
8
7
2
4

Contoh Output

0
1
2
4
7
11
13
15

Note: Soal diambil dari <http://www.spoj.com/problems/BST/>. Soal asli memiliki ukuran input yang jauh lebih besar, sehingga algoritma naïve tidak cukup cepat. Bagi yang ingin memahami BST lebih lanjut, silahkan mengerjakan soal tersebut ☺. Hints dapat dicari di forum SPOJ.