

Started on	Saturday, 24 May 2025, 8:42 PM
State	Finished
Completed on	Saturday, 24 May 2025, 10:25 PM
Time taken	1 hour 42 mins
Overdue	2 mins 18 secs
Marks	300.00/300.00
Grade	10.00 out of 10.00 (100%)

Question **1**

Correct

Mark 100.00 out of 100.00

Time limit	1 s
Memory limit	64 MB

Nama File: bst_range_sum.c

Setelah kemarin mendapatkan 3 macam rute berdasarkan pembacaan traversal (PreOrder, InOrder, dan PostOrder), ketiga Nimons tersebut (Kebin, Pop, Stewart) menyadari bahwa tidak semua node dalam BST merupakan lokasi aman. Berdasarkan informasi intelijen rahasia, hanya node-node yang memiliki key dalam rentang tertentu yang dapat dipercaya sebagai bagian dari jalur penyelamatan Gro. Lokasi yang berada di luar rentang tersebut berpotensi adanya jebakan yang dapat membahayakan misi.

Untuk mempersiapkan strategi dan logistik perjalanan, para Nimons ingin menghitung **total** suplai yang tersedia di semua lokasi yang termasuk **dalam rentang** aman, yaitu **[L, R]** dengan mengimplementasikan **seluruh** fungsi/prosedur [bst_range_sum.h](#).
Anda hanya diminta mengumpulkan `bst_range_sum.c` nya saja. Selain itu, tersedia pula file [boolean.h](#) yang dapat digunakan bila diperlukan dalam implementasi Anda.

Batasan

- $1 \leq \text{info.key} \leq 1000$
- $1 \leq L \leq R \leq 1000$
- Tiap key itu unik

Gunakan [main.c](#) untuk menguji program yang Anda buat.

Keluaran yang diharapkan dari main.c tersebut:

Hasil penjumlahan key node dalam rentang [5, 10] adalah 0
Hasil penjumlahan key node dalam rentang [4, 13] adalah 48

NOTE:

Pastikan setiap output diakhiri oleh **endline** ("**\n**")!

C

 [bst_range_sum.c](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	1	Accepted	0.00 sec, 1.67 MB
2	1	Accepted	0.00 sec, 1.51 MB
3	1	Accepted	0.00 sec, 1.60 MB
4	1	Accepted	0.00 sec, 1.67 MB
5	1	Accepted	0.00 sec, 1.50 MB
6	1	Accepted	0.00 sec, 1.60 MB
7	1	Accepted	0.00 sec, 1.67 MB
8	1	Accepted	0.00 sec, 1.67 MB
9	1	Accepted	0.00 sec, 1.65 MB

No	Score	Verdict	Description
10	1	Accepted	0.00 sec, 1.72 MB
11	9	Accepted	0.00 sec, 1.63 MB
12	9	Accepted	0.00 sec, 1.51 MB
13	9	Accepted	0.00 sec, 1.72 MB
14	9	Accepted	0.00 sec, 1.67 MB
15	9	Accepted	0.00 sec, 1.51 MB
16	9	Accepted	0.00 sec, 1.60 MB
17	9	Accepted	0.00 sec, 1.64 MB
18	9	Accepted	0.00 sec, 1.61 MB
19	9	Accepted	0.00 sec, 1.61 MB
20	9	Accepted	0.00 sec, 1.71 MB

Question **2**

Correct

Mark 100.00 out of 100.00

Time limit	1 s
Memory limit	64 MB

Nama File: bst.c

Kebin diberikan tugas tambahan untuk melengkapi bst.

Bantulah Kebin dengan melengkapi [bst.h](#). Kumpulkan hanya bst.c. Berikut file [boolean.h](#). Untuk mempermudah gunakan BST yang telah diimplementasikan pada saat prapraktikum.

Gunakan [main.c](#) untuk menguji program yang dibuat.

Keluaran yang diharapkan:

Leaf : 2

Depth : 3

Delete : 6

Delete : 2

Delete : 3

Pastikan setiap output diakhiri oleh **endline ("\n")!**

C

 [bst.c](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	25	Accepted	0.00 sec, 1.63 MB
2	25	Accepted	0.00 sec, 1.60 MB
3	25	Accepted	0.00 sec, 1.66 MB
4	25	Accepted	0.00 sec, 1.67 MB

Question **3**

Correct

Mark 100.00 out of 100.00

Time limit	1 s
Memory limit	64 MB

Nama File: bst.c

Setelah Nimons berhasil membuat gadget "Banana Sorting 5000", mereka sangat bangga dengan gadget buatan mereka tersebut. Namun, lagi lagi Dr. Neroifa datang dan mengkritik secara keras gadget buatan Nimons tersebut. Dr. Neroifa mengomentari gadget tersebut dengan julid. "Gadget doang bagus, tapi gabisa diliat isinya 🙄," kata Dr. Neroifa.

Nimons yang sakit hati pun akhirnya berencana untuk mematahkan komentar Dr. Neroifa tersebut. Mereka akhirnya memiliki ide untuk membuat fungsi tambahan yaitu **Beautiful View**. Namun, mereka kebingungan karena ada 3 cara dalam membuat view yang indah, yaitu dengan urutan pre, in, dan post. Oleh karena itu, mereka yang ingin terlihat flawless pun berencana mengimplementasikan ketiga jenis view tersebut.

Bantulah Nimons untuk menyelesaikan ADT ini dengan melengkapi file [bst.h](#). Kumpulkan hanya file **bst.c**. Gunakan file [main.c](#) untuk melakukan pengujian terhadap implementasi yang kalian buat. Output yang diharapkan adalah sebagai berikut.

Preorder:

(50)

Inorder:

(50)

Postorder:

(50)

Preorder:

(50(30)(x))

Inorder:

((30)50(x))

Postorder:

((30)(x)50)

Preorder:

(50(30(20)(40))(70(60)(80)))

Inorder:

(((20)30(40))50((60)70(80)))

Postorder:

(((20)(40)30)((60)(80)70)50)

Lampiran:

[boolean.h](#)

Notes:

- Jika node merupakan leaf node, tampilkan hanya value dari node tersebut, tidak perlu menampilkan children nya.
- Jika node memiliki minimal 1 children, tampilkan kedua children nya. (sesuai contoh main.c)

C

 [bst.c](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	10	Accepted	0.00 sec, 1.56 MB
2	10	Accepted	0.00 sec, 1.65 MB
3	10	Accepted	0.00 sec, 1.61 MB
4	10	Accepted	0.00 sec, 1.59 MB
5	10	Accepted	0.00 sec, 1.59 MB
6	10	Accepted	0.00 sec, 1.67 MB
7	10	Accepted	0.00 sec, 1.67 MB
8	10	Accepted	0.00 sec, 1.65 MB
9	10	Accepted	0.00 sec, 1.54 MB
10	10	Accepted	0.00 sec, 1.67 MB

[◀ Praktikum 9 - K3 & K4](#)

Jump to...