

Started on	Thursday, 12 December 2024, 8:46 PM
State	Finished
Completed on	Thursday, 12 December 2024, 10:07 PM
Time taken	1 hour 20 mins
Grade	300.00 out of 300.00 (100%)

Time limit	1 s
Memory limit	64 MB

Lengkapi ADT Pohon Biner yang telah dibuat sebagai tugas pra-praktikum dengan beberapa fungsi/prosedur yang diberikan dalam file [bintree.h](#) yaitu:

Tambahan operasi-operasi:

- boolean isEqual(BinTree p1, BinTree p2);
- int getMaximumDepth(BinTree p);
- BinTree createBinSearchTreeFromSortedArray(EIType* arr, EIType start, EIType end);
- int countNodes(BinTree p);
- void printPathToElement(BinTree p, EIType target);

Kumpulkan file **bintree.c**.



[bintree.c](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	5	Accepted	0.00 sec, 1.67 MB
2	5	Accepted	0.00 sec, 1.62 MB
3	5	Accepted	0.00 sec, 1.67 MB
4	5	Accepted	0.00 sec, 1.55 MB
5	5	Accepted	0.00 sec, 1.56 MB
6	5	Accepted	0.00 sec, 1.57 MB
7	5	Accepted	0.00 sec, 1.65 MB
8	5	Accepted	0.00 sec, 1.61 MB
9	5	Accepted	0.00 sec, 1.63 MB
10	5	Accepted	0.00 sec, 1.56 MB
11	5	Accepted	0.00 sec, 1.61 MB
12	5	Accepted	0.00 sec, 1.63 MB
13	5	Accepted	0.00 sec, 1.64 MB
14	5	Accepted	0.00 sec, 1.64 MB

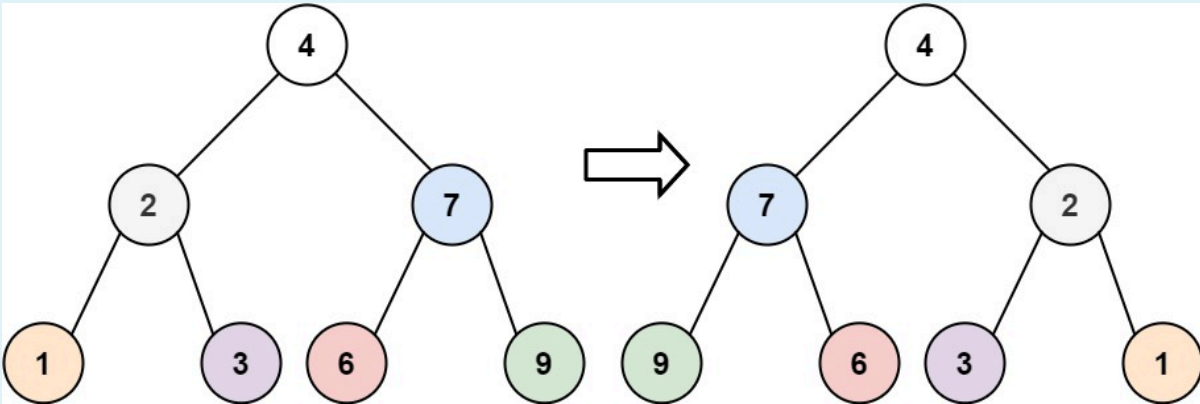
No	Score	Verdict	Description
15	5	Accepted	0.00 sec, 1.64 MB
16	5	Accepted	0.00 sec, 1.60 MB
17	5	Accepted	0.00 sec, 1.63 MB
18	15	Accepted	0.00 sec, 1.61 MB

Time limit	1 s
Memory limit	64 MB

Invert

Deskripsi

Diberikan sebuah pohon biner **T**. Balikkan subtree kiri dan kanan pada setiap node dari pohon **T**. Pohon bisa kosong. Jika pohon kosong, *do nothing*.



Implementasi fungsi **invertTree** menggunakan template [invert_template.c](#)

Catatan

- Kumpulkan dalam file bernama **invert.c**

C

[invert.c](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	20	Accepted	0.00 sec, 1.64 MB
2	20	Accepted	0.00 sec, 1.64 MB
3	20	Accepted	0.00 sec, 1.64 MB
4	20	Accepted	0.00 sec, 1.60 MB
5	20	Accepted	0.00 sec, 1.63 MB

Time limit	1 s
Memory limit	64 MB

Pencurry

Deskripsi

Agen Purry sedang melaksanakan sebuah misi terakhirnya yaitu pencurian yang akan dikenang sepanjang masa. Pencurian ini akan menentukan hidup dan mati OWCA sehingga Purry tidak boleh melakukan kegagalan dalam misi ini.

Pada misi pencurian ini, Purry dan kamu (asisten Purry) harus mengambil harta dari markas organisasi musuh sebanyak-banyaknya. Untuk membuatmu bingung, musuh merancang markas mereka menjadi sebuah labirin yang membingungkan. Namun, kamu telah mendapat info A1 bahwa markas ini tidak serumit labirin yang dikira dan bisa dimodelkan menjadi Binary Tree.

Dengan model Binary Tree, **root** dari model adalah **pintu masuk**. Lalu, setiap ruangan dalam markas musuh dapat dimodelkan menjadi node. **Jumlah harta** dalam setiap ruangan adalah **info** dari node tersebut. Terakhir, left dan right adalah node tetangga **yang dapat diakses** dari ruangan tersebut.

Namun, organisasi musuh tidaklah bodoh. Agar mudah mendeteksi adanya kebobolan, mereka membuat mekanisme sistem pertahanan. Namun, info A1-mu mengatakan bahwa sistem pertahanan mereka memiliki kelemahan yaitu sistem hanya akan **aktif** apabila terjadi pencurian di **2 ruangan bersebelahan**.

Dengan pengetahuan ini dan peta mengenai jumlah harta di setiap ruangan, kamu dan Purry dapat menentukan **jumlah harta maksimum** yang bisa kalian dapatkan. Tapi, Purry masih disibukkan dengan urusan administrasi lainnya sehingga kamu ditugaskan untuk membuat rencananya. Bantulah Purry menentukan **jumlah harta maksimum** yang bisa kalian dapatkan.

Kumpulkan file dengan fungsi mencurry yang menyelesaikan permasalahan ini dengan nama file [pencurry.c](#) (Gunakan file yang dilampirkan sebagai template)

Format Parameter

Sebuah Binary Tree T yang merupakan peta dari markas musuh.

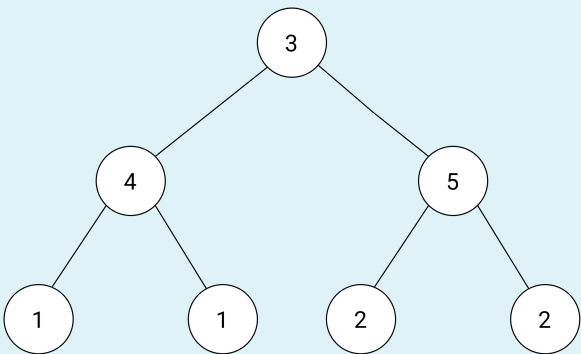
Format Kembalian

Sebuah integer yang merupakan jumlah harta maksimum yang bisa kalian dapatkan tanpa mengambil dari 2 ruangan bersebelahan.

Catatan Penting

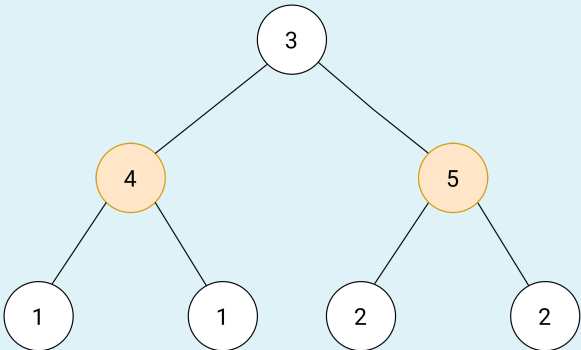
Diperbolehkan membuat fungsi bantuan baru.

Contoh

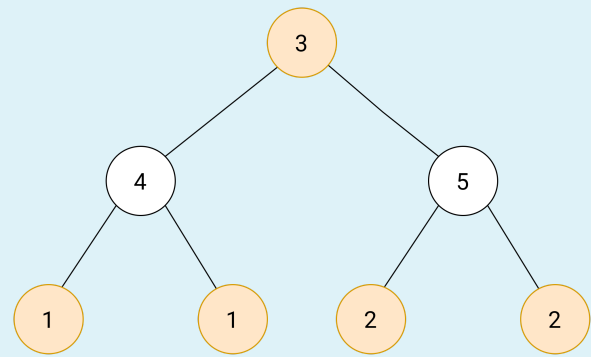


Berikut adalah contoh sebuah Binary Tree. Pada masukan ini, hasil kembalian program seharusnya 9. Berikut adalah contoh konfigurasi yang bisa menghasilkan angka tersebut.

Konfigurasi 1:



Konfigurasi 2:



Penjelasan

Untuk konfigurasi 1, dapat dilihat bahwa tidak ada lagi node yang dapat diambil. Oleh karena itu, keluarannya adalah:
 $4 + 5 = 9$

Untuk konfigurasi 2, dapat dilihat bahwa tidak ada lagi node yang dapat diambil. Oleh karena itu, keluarannya adalah:
 $3 + 1 + 1 + 2 + 2 = 9$



Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	5	Accepted	0.00 sec, 1.72 MB
2	5	Accepted	0.00 sec, 1.64 MB
3	5	Accepted	0.00 sec, 1.57 MB
4	5	Accepted	0.00 sec, 1.51 MB
5	5	Accepted	0.00 sec, 1.57 MB
6	5	Accepted	0.00 sec, 1.57 MB
7	5	Accepted	0.00 sec, 1.62 MB
8	5	Accepted	0.00 sec, 1.63 MB
9	5	Accepted	0.00 sec, 1.68 MB
10	5	Accepted	0.00 sec, 1.68 MB
11	5	Accepted	0.00 sec, 1.64 MB
12	5	Accepted	0.00 sec, 1.67 MB
13	5	Accepted	0.00 sec, 1.74 MB
14	5	Accepted	0.00 sec, 1.70 MB
15	5	Accepted	0.00 sec, 1.66 MB
16	5	Accepted	0.00 sec, 1.94 MB
17	5	Accepted	0.00 sec, 2.11 MB
18	5	Accepted	0.00 sec, 2.28 MB

No	Score	Verdict	Description
19	5	Accepted	0.00 sec, 2.31 MB
20	5	Accepted	0.00 sec, 2.36 MB

[◀ Pohon Biner - Praktikum](#)

Jump to...

⌵