

پیمایش

1- پیمایش سطحی (Level Order) یک درخت دودویی به صورت زیر داده شده است.

$$\text{Level Order} = [A, B, C, D, E, F, G, H, I]$$

الف) درخت دودویی معادل این پیمایش را رسم کنید.

ب) پس از رسم پیمایش‌های پیشوندی (Preorder)، میانوندی (Inorder) و پسوندی (Postorder) این درخت را بنویسید.

ج) آیا می‌توانید نود F را با استفاده از (DFS) پیدا کنید؟ مسیر جستجو را بنویسید.

د) آیا می‌توانید نود H را با استفاده از (BFS) پیدا کنید؟ مسیر جستجو را بنویسید.

ه) این درخت را به صورت ترتیبی از ریشه بنویسید. (با استفاده از الگوریتم DFS و BFS)

ز) عمق و ارتفاع نودهای A، H و G را به دست آورید.

پیاده‌سازی درخت دودویی

به تمامی قسمت‌های زیر پاسخ کامل دهید.

الف) چگونه می‌توان یک **درخت دودویی** را با استفاده از آرایه پیاده‌سازی کرد؟ برای هر گره در آرایه، نحوه محاسبه نمایه (Index) فرزندان چپ و راست و والد را توضیح دهید.

ب) چگونه می‌توان یک درخت دودویی را با استفاده از لیست پیوندی پیاده‌سازی کرد؟ نحوه ذخیره‌سازی گره‌ها و ارتباطات بین آن‌ها را توضیح دهید.

ج) این دو پیاده‌سازی را از لحاظ پیچیدگی زمانی مقایسه کنید.

د) در چه شرایطی استفاده از پیاده‌سازی درخت دودویی با آرایه توصیه می‌شود و در چه شرایطی پیاده‌سازی با لیست پیوندی مناسب‌تر است؟ مثال‌هایی از کاربردهای واقعی برای هر یک از این روش‌ها ارائه دهید.

شبه کد درخت دودویی

شبه کد پیاده سازی **درخت دودویی** با استفاده از **آرایه** و **لیست پیوندی** را بنویسید.

زیر درخت

فرض کنید پیمایش‌های میانوندی (inorder)، پیشوندی (preorder) و ریشه دو درخت T_1 و T_2 موجود است.

الف) الگوریتمی طراحی کنید که بررسی کند که آیا درخت T_2 یک زیر درخت از T_1 است یا خیر.

ب) شبه‌کد الگوریتم خود را بنویسید.

ج) الگوریتم خود را برای ورودی‌های زیر آزمایش (trace) کنید.

Inorder_ T_1 = 1,3,6,2,8,4,5

Inorder_ T_2 = 1,3,6,2

Preorder_ T_1 = 8,6,1,3,2,4,5

Preorder_ T_2 = 6,1,3,2

تبدیل درخت کامل به باینری



درخت‌های عمومی (General Tree) ساختاری هستند که هر گره می‌تواند چندین فرزند داشته باشد. از طرفی درخت‌های دودویی (Binary Tree) ساختاری هستند که هر گره حداکثر دو فرزند (چپ و راست) می‌تواند داشته باشد. شما باید الگوریتمی طراحی کنید که یک درخت عمومی را به یک درخت دودویی تبدیل کند. سپس شبه کد آنرا بنویسید.

مسیریابی

- محدودیت زمان: 1 ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت



به شما ریشه (root) یک درخت دودویی و یک عدد صحیح به نام `targetSum` داده شده است. تمامی مسیرهای از ریشه تا برگ که جمع مقادیر گره‌های مسیر برابر با `targetSum` باشد را بازگردانید. هر مسیر باید به صورت یک

لیست از مقادیر گره‌ها باشد.

یک مسیر از ریشه تا برگ مسیری است که از ریشه شروع و به هر گره برگ ختم می‌شود. یک گره برگ، گره‌ای است که فرزندی ندارد.

ورودی

ورودی شامل 4 خط است. در خط اول، عدد n که تعداد نودهای درخت است به شما داده می‌شود. در خط دوم و سوم به ترتیب پیمایش‌های Pre Order و In Order درخت به شما داده می‌شود. هر خط شامل n عدد است که به وسیله فاصله از هم جدا شده‌اند. در خط چهارم، عدد $targetSum$ داده می‌شود.

$$0 \leq n \leq 5000$$

خروجی

در خروجی در هر خط مسیرهای امکان پذیر باید نمایش داده شود.

مثال

در اینجا چند نمونه برای فهم بهتر صورت سوال و قالب ورودی و خروجی تست‌ها داده می‌شود.

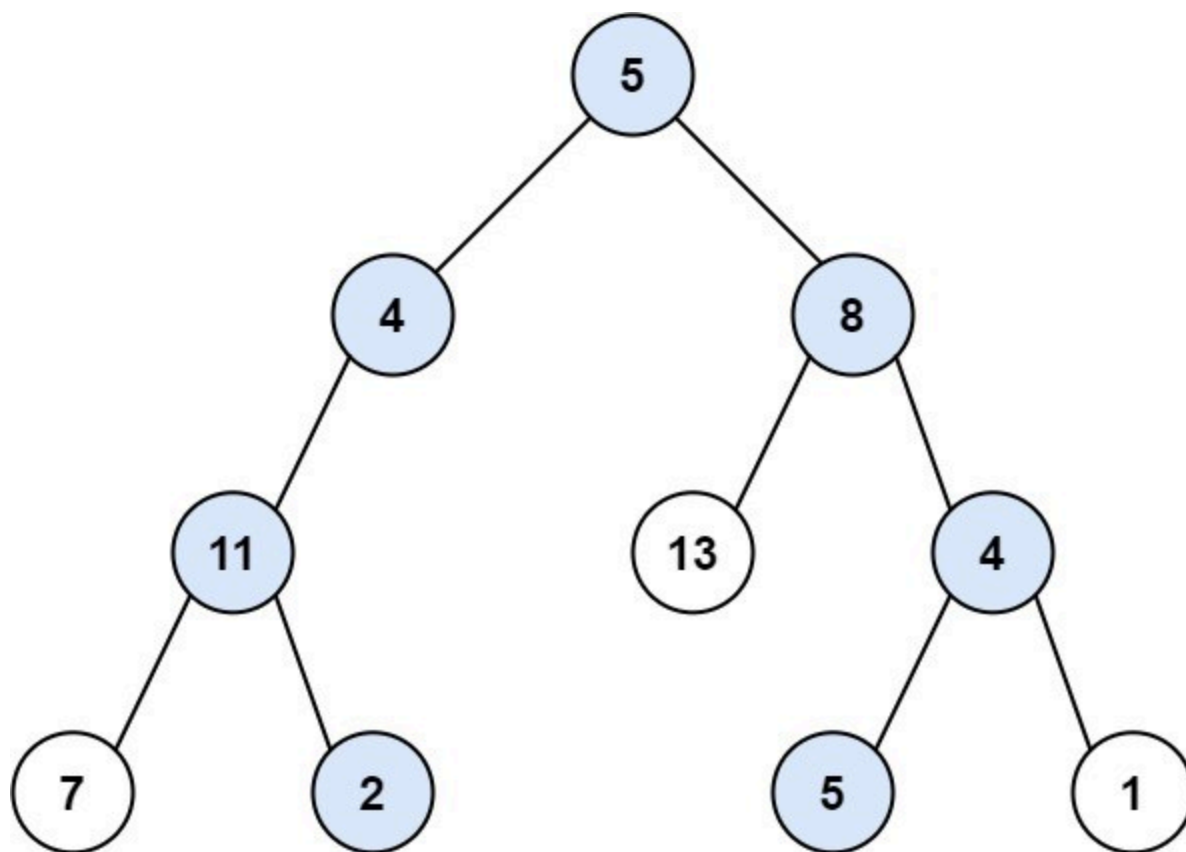
ورودی نمونه ۱

```
10
5 4 11 7 2 8 13 4 5 1
7 11 2 4 5 13 8 5 4 1
22
```

خروجی نمونه ۱

```
5 4 11 2
5 8 4 5
```

توضیح نمونه:



دو مسیر وجود دارد که جمع آنها برابر با 22 است

$$5 + 4 + 11 + 2 = 22$$

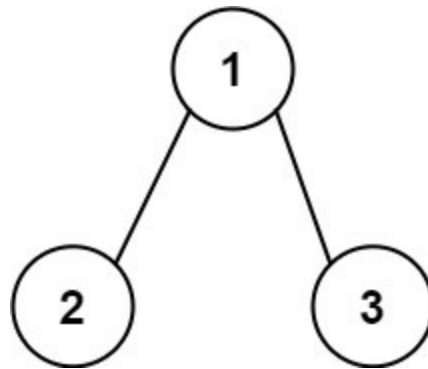
$$5 + 8 + 4 + 5 = 22$$

ورودی نمونه ۲

3
1 2 3
2 1 3
5

خروجی نمونه ۲

توضیح نمونه:



هیچ مسیری وجود ندارد که مجموع آن 5 باشد.

مانیتورینگ

- محدودیت زمان: ۱ ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت



به شما ریشه یک درخت دودویی داده شده است. ما دوربین‌هایی را بر روی نودهای درخت نصب می‌کنیم که هر دوربین می‌تواند والد، خودش، و فرزندان مستقیم خود را مانیتور کند. حداقل تعداد دوربین‌های مورد نیاز برای پوشش تمامی نودهای درخت را بازگردانید.

در این سوال فقط ساختار درخت مهم است و مقدار هر گره درخت برابر با صفر است.

ورودی

در خط اول، عدد n که تعداد نودهای درخت است به شما داده می‌شود. در خط بعدی پیمایش `levelorder` درخت داده می‌شود. منظور از 0 یعنی نود وجود دارد و منظور از -1 به معنای وجود نداشتن نود است.

$$0 \leq n \leq 1000$$

خروجی

یک عدد بازگردانده می‌شود که نشان دهنده حداقل تعداد دوربین‌ها است.

مثال

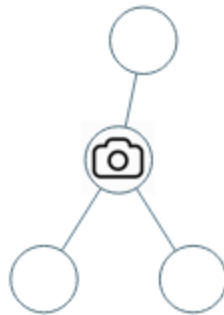
ورودی نمونه ۱

```
4
0 0 -1 0 0
```

خروجی نمونه ۱

```
1
```

توضیح نمونه:



یک دوربین برای پوشش تمامی نودها لازم است.

ورودی نمونه ۲

5

0 0 -1 0 -1 0 -1 -1 0

خروجی نمونه ۲

2

توضیح نمونه:



حداقل دو دوربین برای پوشش تمامی نودها لازم است.