

Post-Order -> Pre-Order : ( $N_3=3$ ) < 5

ابتدا المنت های یک BST که به روش Post-Order پیمایش و مرتب شده اند ، یکی یکی وارد می کنیم و پس از وارد کردن آخرین المنت end را برای اتمام دادن ورودی وارد می کنیم.

داده ها وارد شده در یک ArrayList ذخیره شده اند که در ابتدا نیازی به مشخص کردن تعداد نباشد چون قرار است برای آرایه جدیدی که قرار است جواب باشد داده ها را از آخر به اول وارد کنیم برای کار کردن بهتر با index ها ، داده ها را به یک آرایه وارد می کنیم.

از یک حلقه استفاده می کنیم و بزرگترین و کوچک ترین المنت را مشخص می کنیم.

یک آرایه دیگر هم اندازه آرایه اول ایجاد می کنیم تا حاصل پیمایش جدید (Pre-Order) را داخل آن قرار دهیم.

یک تابع به نام PostToPre داریم المنت های آرایه جدید (پیمایش Pre-Order) را مشخص می کند.

فرمت این تابع به این شکل است:

PostToPre(int f, int minValue, int maxValue, int preIndex, int[] postorder, int[] preorder)

f : index جایی زیردرخت در حالت Post-Order تمام می شود (اینجا وقتی به آخرین المنت زیردرخت در حالت Post-Order می رسم که  $f = -1$  شود).

minValue: کوچکترین المنت زیردرخت را مشخص می کند.

maxValue: بزرگترین المنت زیر درخت را مشخص می کند.

preIndex: index خانه ای از آرایه است که حاصل قرار است داخل آن نوشته شود.

postorder: پیمایش درخت به صورت Post-Order

preorder: پیمایش درخت به صورت Pre-Order

الگوریتم این تابع به این شکل است که ابتدا پیمایش Post-Order درخت جست و جوی باینری (BST) را دریافت می کند.

در Post-Order ترتیب به صورت (left, right, root) است . همچنین چون است BST ترتیب (left < root < right) را هم داریم.

چون Post-Order است اولین عنصر از سمت چپ root است از آن به بعد به چپ حرکت می کنیم تا جایی که عناصر از root بزرگتر باشند، از root به بعد تا آن عنصر، زیردرخت راست را تشکیل می دهد و از آن به بعد تا ابتدای آرایه زیر درخت چپ را تشکیل می دهد.

ترتیب در Pre-Order به صورت (root, left, right) است .

می توانیم preorder را از آخر به اول پر کنیم یعنی اول زیردرخت راست ، بعد زیردرخت چپ و سپس root را قرار دهیم.

برای این کار در postorder ابتدا در زیردرخت راست به صورت recursive حرکت می کنیم . دوباره به صورت

(left, right, root) آن را به زیر درخت چپ و راست و root تقسیم کرده تا به پایان زیر رشته راست برسیم و بعد راست ترین، چپ ترین و root را در preorder از راست به چپ می نویسم (در تمام زیر درخت های ساخته شده این کار را به صورت بازگشتی تکرار می کنیم و هربار راست ترن زیردرخت را برای پیمایش انتخاب می کنیم چون می خواهیم preorder (آخر به اول) قرار دهیم).

وقتی زیر درخت راست تمام شد ، به زیردرخت چپ می رویم و همان روندی را که در زیردرخت راست طی کردیم به صورت recursive طی کرده تا به آخر postorder برسیم و در آخر این عملیات بازگشتی هم root در جایگاه اولین عنصر preorder قرار می گیرید.

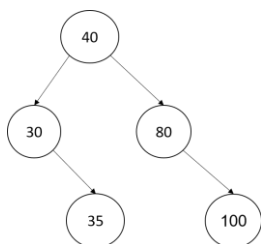
درپایان ، پیمایش Pre-Order که در آرایه preorder ذخیره شده را چاپ می کنیم.

مثال 1:

Input : 35 30 100 80 40

Output : 40 30 35 80 100

35	30	100	80	40
left		right		root



100	80
right	root

100
root

				100
--	--	--	--	-----

			80	100
--	--	--	----	-----

35	30
right	root

35
root

		35	80	100
--	--	----	----	-----

	30	35	80	100
--	----	----	----	-----

40
root

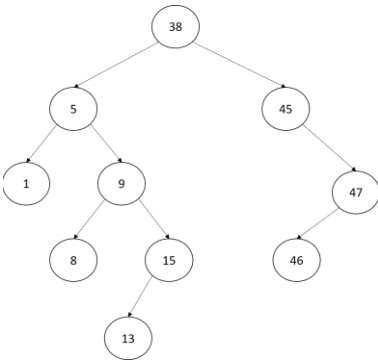
40	30	35	80	100
----	----	----	----	-----

مثال 2:

Input: 1 8 13 15 9 5 46 47 45 38

Output: 38 5 1 9 8 15 13 45 47 46

1	8	13	15	9	5	46	47	45	38
left						right		root	



46	47	45
right		root

46	47
left	root

46
root

									46
--	--	--	--	--	--	--	--	--	----

								47	46
--	--	--	--	--	--	--	--	----	----

							45	47	46
--	--	--	--	--	--	--	----	----	----

1	8	13	15	9	5
left	right				root

8	13	15	9
left	right		root

13	15
left	root

13
root

						13	45	47	46
--	--	--	--	--	--	----	----	----	----

					15	13	45	47	46
--	--	--	--	--	----	----	----	----	----

8
left

				8	15	13	45	47	46
--	--	--	--	---	----	----	----	----	----

9
root

			9	8	15	13	45	47	46
--	--	--	---	---	----	----	----	----	----

1
root

		1	9	8	15	13	45	47	46
--	--	---	---	---	----	----	----	----	----

5
root

	5	1	9	8	15	13	45	47	46
--	---	---	---	---	----	----	----	----	----

38
root

38	5	1	9	8	15	13	45	47	46
----	---	---	---	---	----	----	----	----	----

