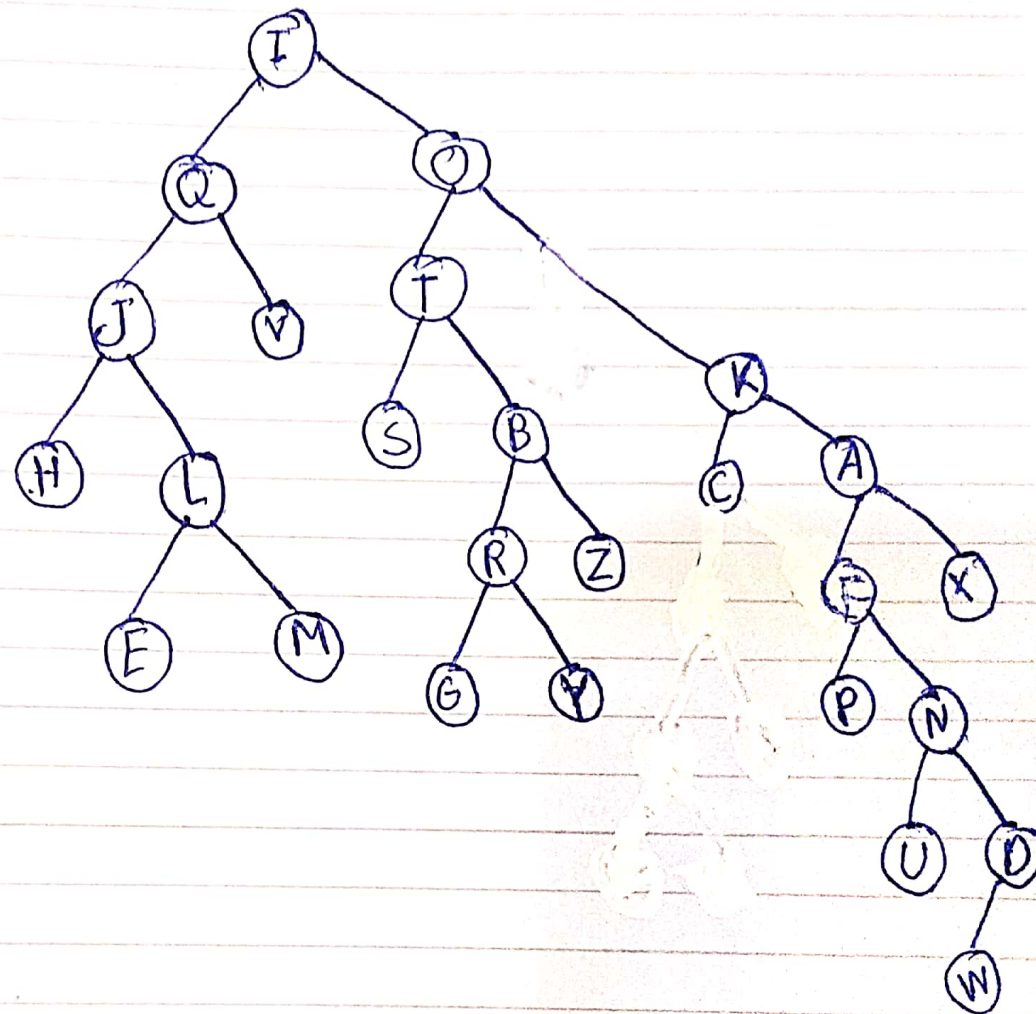


97521252

آریان حبیری

1(5) همواره اولین عنصر از  $pre-order$  به ما  $root$  را نشان می‌دهد. و همچنین آخرین عنصر از  $post order$  هم همان را نشان می‌دهد. عنصر بعدی در  $pre-order$  هم زیرشاخه سمت چپ و عنصر قبلی در  $post order$  هم زیرشاخه سمت راست را نشان می‌دهد.

در این جا به سادگی تابع را با ورودی داده شده برای  $pre order$  می‌خوانی می‌کنیم. سپس  $I$  را در نظر گرفته و برای نیمه چپ آن همین تابع را با  $pre-order$  از  $Q$  به بعد و  $post-order$  را از  $Q$  به قبل فراخوانی می‌کنیم. همچنین برای نیمه راست همین تابع را با  $pre-order$  از  $O$  به بعد و یا  $post-order$  از  $O$  به قبل فراخوانی می‌کنیم. و همین روند بازگشتی را در هر نیمه انجام می‌دهیم و به این ترتیب همه عناصر در جای خود قرار می‌گیرند. انقدر کار را ادامه می‌دهیم تا عضری انتخاب شود که قبلاً شده بود. یعنی باید در هر بار فراخوانی تابع چک کنیم که عنصر آخر از  $post order$  (و عنصر اول از  $pre order$ ) که می‌خواهیم به درخت اضافه اش کنیم قبلاً اضافه نشده باشد، اگر شده باشد دیگر کار تابع در آن شاخه تمام است. و شکل نهایی درخت اینطور خواهد بود:



2. در هر مرحله باید یک دور تمام رئوس اضافه شده چک شود و همچنین دوبار برای سمت

چپ و راست تابع فراخوانی شود.

$$T(n) = 2T\left(\frac{n}{2}\right) + T(n) \rightarrow O(n) = n^{\log_2 2} \cdot \log_2 n = O(n \log n)$$