



دانشکده علوم ریاضی و آمار



نیمسال اول ۱۴۰۰-۱۴۰۱

مدرس: دکتر مجتبی رفیعی

ساختمان داده‌ها و الگوریتم‌ها

جلسه ۲۶

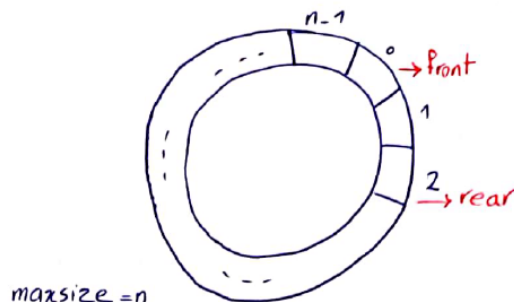
نگارنده: فاطمه هاشمیان

۲۶ آبان ۱۴۰۰

فهرست مطالب

۱	۱ صف حلقوی
۲	۲ درخت‌ها
۶	۳ داده ساختار درخت

۱ صف حلقوی



* $front$ به ابتدای صف (Queue) اشاره می‌کند،

* $rear$ به انتهای صف (Queue) اشاره می‌کند؛ جایی که داده جدید باید درج شود (انتهای صف)

* برای بروز کردن $front$ (حذف کردن عنصر از ابتدای صف):

$$front = (front + 1) \bmod n$$

* برای بروز کردن $rear$ (اضافه کردن عنصر به انتهای صف):

$$rear = (rear + 1) \bmod n$$

* خالی بودن و پر بودن صف بوسیله شرط $rear = front$ چک می‌شود، اما برای تفکیک این دو:

۱- نگه داشتن یک متغیر برای اندازه صف،

۲- یک خانه اضافه‌تری را در نظر بگیریم (به این منظور که نمی‌تواند مقدار بگیرد)، مثلاً در $n-1$ عنصری نگذاریم. بنابراین، اگر

$real = front$ آنگاه صف خالی است. و اگر $real = (real + 1) \bmod n$ آنگاه صف پر است.

تمرین: شبه‌کد مربوط به هر یک از عملیات زیر را برای نمایش حلقوی (هر دو رویکرد نگهداری اندازه صف و عنصر اضافی) صف بنویسید.
 $is\ full(Q)$ یک پرسمان بازیابی است که صف Q را بعنوان ورودی دریافت می‌کند و تعیین می‌کند که آیا صف Q پر است یا نه؟ در صورت پر بودن $True$ و در غیر این‌صورت $False$ را برمی‌گرداند.

$is\ Empty(Q)$: یک پرسمان بازیابی است که صف Q را بعنوان ورودی دریافت می‌کند که آیا Q خالی است یا خیر (در صورت خالی بودن $True$ و در غیر این‌صورت $False$ را برمی‌گرداند).

* $EnQueue(Q, x)$: شبه تعریف ارائه شده برای نمایش ۱ می‌باشد.

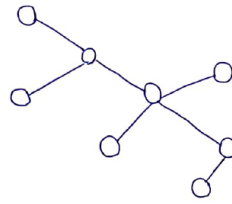
* $DeQueue(Q)$: شبه تعریف ارائه شده برای نمایش ۱ می‌باشد.

۲ درخت‌ها

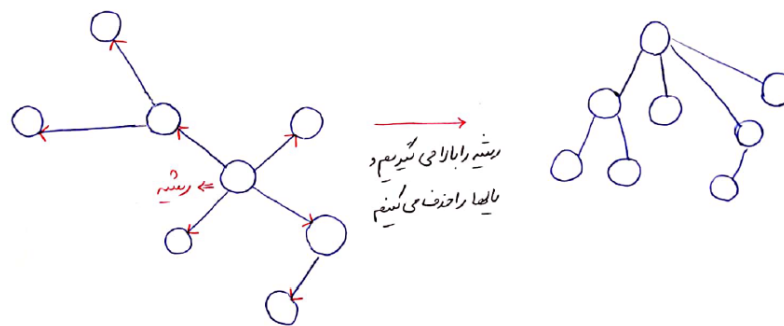
در این بخش، برخی از مفاهیم و تعارف مربوط به درخت‌ها جهت یادآوری آورده شده است.

مفاهیم و تعاریف مربوط به درخت‌ها:

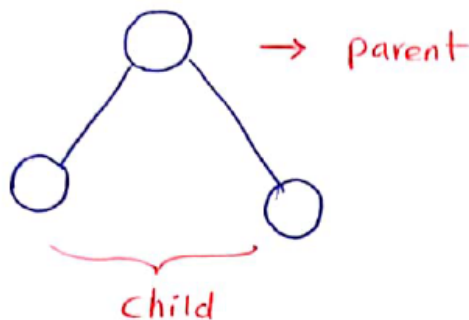
* درخت آزاد ($Free\ tree$): گراف همبندی است که دور ندارد مثل



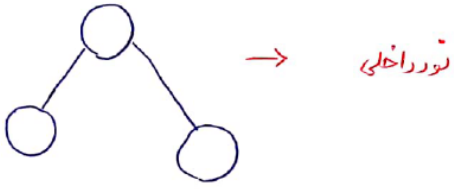
*درخت ریشه‌دار (Rooted tree) درختی است که در آن یک گره به عنوان ریشه لحاظ می‌شود و یالها طوری جهت‌دار می‌شوند که هر گره غیر ریشه، درجه ورودی ۱ و ریشه درجه ورودی ۰ دارد، مثال:



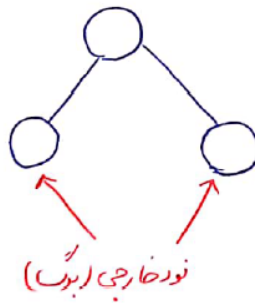
نکته: به ازای تعداد گره‌های درخت آزاد، می‌توان درخت ریشه‌دار داشت. لازم به ذکر است هر گره را به عنوان ریشه در نظر بگیریم یک درخت یکتا به دست می‌آید. تعاریفی که در ادامه می‌آید تماماً روی درخت‌های ریشه‌دار در نظر گرفته شده است. *رابطه پدر-فرزندی (Parent - child): بین هر یال درخت می‌توان این رابطه را تعریف کرد با توجه به یال‌های متصل به نود به نودی که نزدیک به نود ریشه است نود پدر و نود دیگر نود فرزند نام دارد.



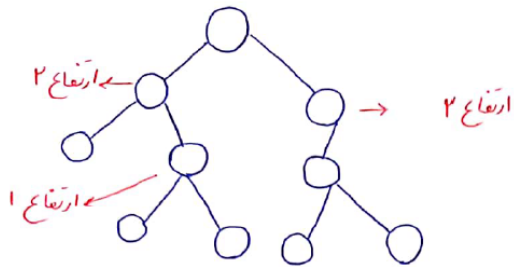
*نودهای داخلی (Internal nodes): به نودهایی که یک یا بیشتر نود فرزند دارند نود داخلی گفته می‌شود.



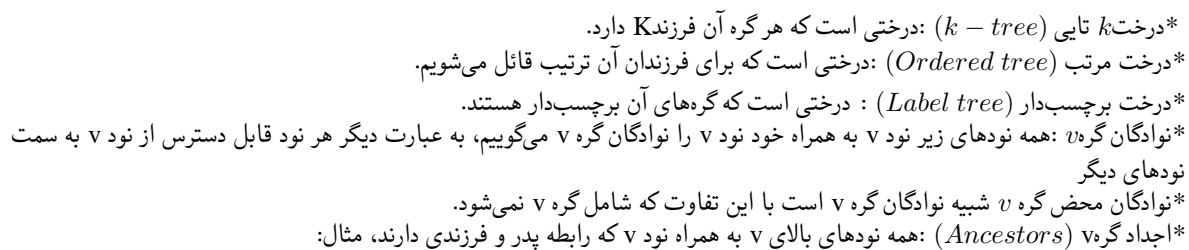
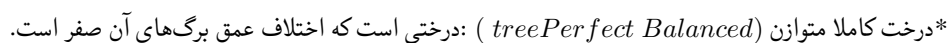
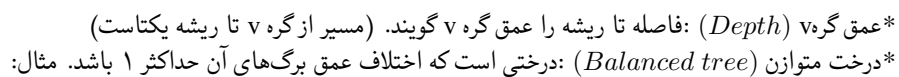
*نودهای خارجی یا نودهای برگ (*Leaf nodes*): به نودهایی که نود فرزند ندارند گفته می‌شود.

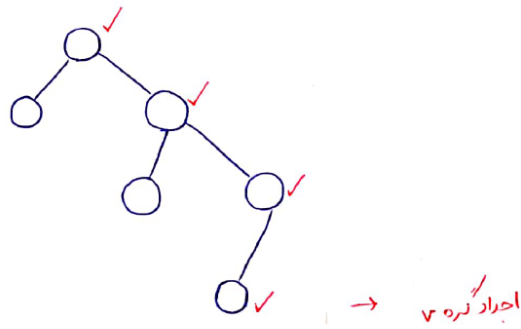


*ارتفاع گره v (*Height*): طولانی‌ترین مسیر از گره v به برگ را می‌گوییم.



*ارتفاع درخت (*Tree height*): ارتفاع گره ریشه در درخت را می‌گوییم.

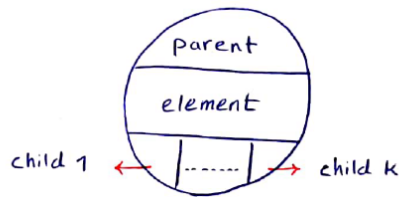




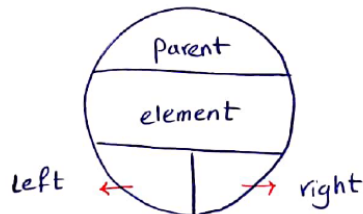
اجزای محض گره v ، شبیه اجزای گره v است با این تفاوت که شامل گره v نمی‌شود.
 * جنگل (Forest): یک مجموعه از درخت‌های مجزا را جنگل گوئیم.

۳ داده ساختار درخت

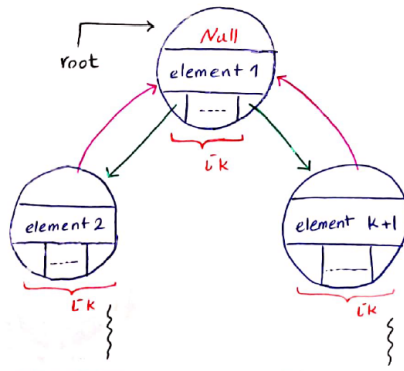
* داده ساختار درخت (Data - structure - tree):
 یک داده ساختار برای نگهداری مجموعه‌های پویاست که عناصر آن در گره‌های یک درخت قرار می‌گیرند.
 * نمایش گرافیکی هر نود درخت k -تایی:



* نمایش گرافیکی هر نود درخت ۲-تایی (دودویی):



* نمایش گرافیکی درخت k -تایی:



برخی نکات:

اگر $T.root.parent = Null$ آنگاه x گره ریشه است (اگر درخت T تهی نباشد).

اگر $T.root = Null$ باشد آنگاه درخت T تهی است.

اگر $x.left = Null$ و $x.right = Null$ باشد آنگاه x در یک درخت دودویی یک گره برگ است، بطور مشابه اگر $x.child_1 = Null$ و $x.child_k = Null...$ در یک درخت k -تایی باشد آنگاه x یک گره برگ است.

نکته: در یک درخت k -تایی ممکن است اکثر اشاره‌گرها به نودهای فرزند $Null$ باشند پس حافظه‌ی زیادی اتلاف می‌شود، در شرایط خاص مثل درخت k -تایی پر و یا متوازن چندان مهم نیست چرا که نشانگر $Null$ اندکی وجود دارد. خوشبختانه روشی وجود دارد که در حالت کلی جلوی این اتلاف را می‌گیرد.