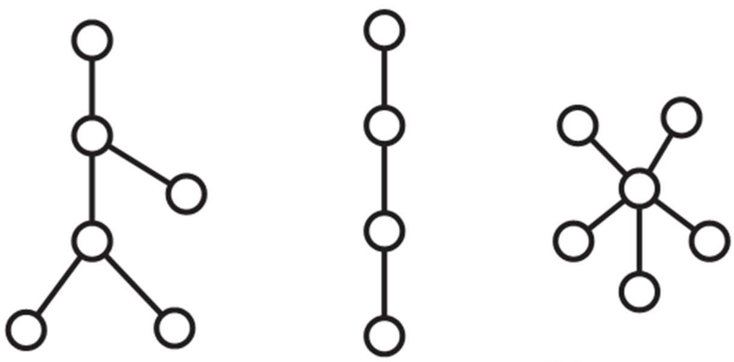
درخت ها و الگوریتم های DFS و BFS

درس: نظریه الگوریتمی گراف

گروه علوم کامپیوتر دانشگاه شهید بهشتی

تعریف‌ها و نتایج اولیه

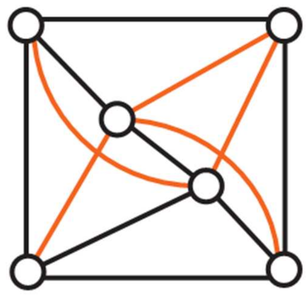
- درخت یک گراف همبند بدون دور است.
- جنگل یک گراف بدون دور است. پس هر مولفه همبندی جنگل ، درخت است.



- هر راس درجه 1 در ...

تعریف‌ها و نتایج اولیه

- یک درخت فراگیر از گراف G یک زیردرخت فراگیر از آن است که درخت باشد.



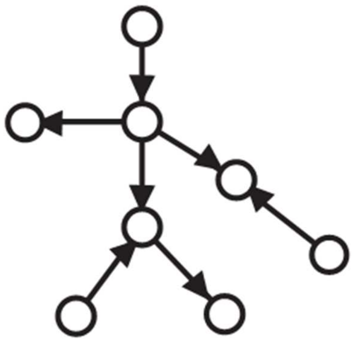
- درخت با یک راس را درخت بدیهی می‌نامیم.

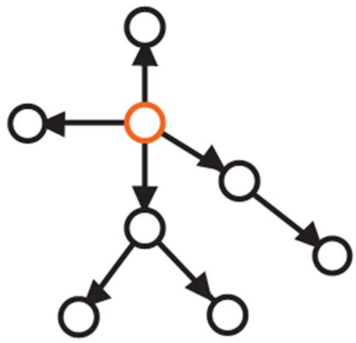
تعریف‌ها و نتایج اولیه

- قضیه: درخت T دارای n راس و $n-1$ یال است.
- قضیه: بین هر دو راس از درخت دقیقا یک مسیر وجود دارد.
- نتیجه: هر یال درخت یک پل است.
- قضیه: هر درخت غیر بدیهی دارای حداقل 2 برگ است.
- قضیه: اگر بزرگترین درجه راسی درخت T برابر با Δ باشد ، آنگاه T دارای حداقل Δ برگ است.

درخت ریشه‌دار

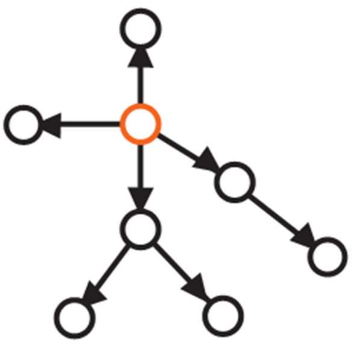
- درخت جهتدار T گراف جهتداری است که گراف زمینه آن درخت باشد.
- درخت ریشه‌دار T درخت جهتداری است که راسی مانند r به نام ریشه داشته باشد به طوری که از ریشه به هر راس دیگر مسیر جهتداری وجود داشته باشد.

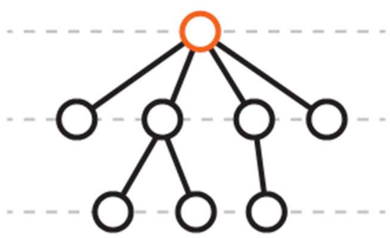




درخت ریشه‌دار

- اگر T یک درخت ریشه دار باشد ، معمول است T طوری رسم شود که ریشه در بالاترین سطح (سطح صفر)، راس‌های مجاور آن در سطح یک و به همین صورت راس‌های مجاور راس‌های هر سطح i در سطح $i+1$ قرار گیرند. در این صورت جهت کمان ها در نمایش حذف می‌شود.





- بزرگترین سطح

درخت ریشه‌دار

- قضیه: درخت جهتدار T ریشه دار است اگر و تنها اگر T شامل راسی مانند r باشد به طوری که $id(r) = 0$ و برای هر راس دیگر u داشته باشیم $id(u) = 1$

ایده اثبات: اگر T درخت ریشه دار باشد ، حکم به وضوح برقرار است.

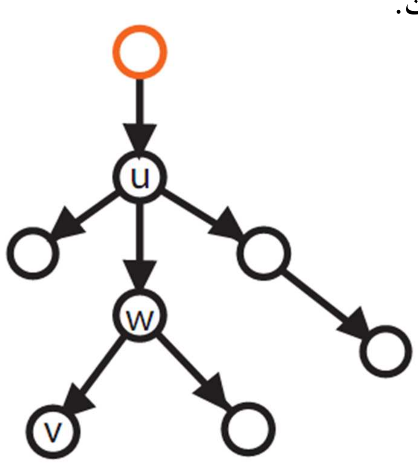
فرض کنید T درخت جهتدار با شرط داده شده باشد. یک راس دلخواه u انتخاب کنید. $id(u) = 1$ ، پس کمان ورودی (v,u) وجود دارد. اگر $v = r$ مساله حل شده است. در غیر این صورت v هم یک کمان ورودی دارد. با ادامه این روند مسیری جهتدار از r به u تعیین می‌شود.

درخت ریشه‌دار

- در درخت ریشه دار T ،

اگر کمان (w,v) وجود داشته باشد ، v فرزند w و w پدر v است.

اگر مسیر جهتداری از u به v وجود داشته باشد ، u جد v و V نوه u است.



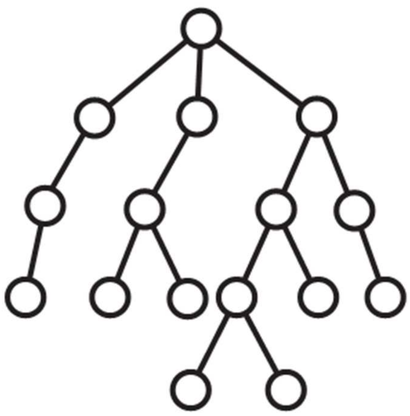
زیردرخت ریشه‌داری که از راس u و همه نوادگان آن تشکیل می‌شود ، زیردرخت ماکسیمال T با ریشه u نام دارد و با نماد $T(u)$ نشان داده می‌شود.

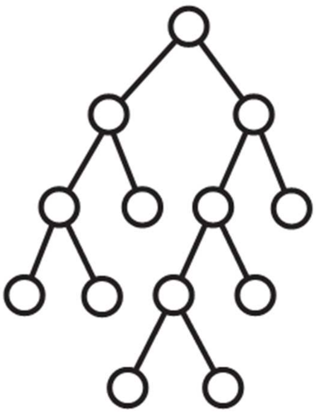
درخت ریشه‌دار

- درخت ریشه دار T را m-تایی می‌نامیم هرگاه هر راس آن حداکثر m فرزند داشته باشد.
- درخت m-تایی را تام می‌نامیم هرگاه هر راس m یا صفر فرزند داشته باشد.
- درخت m-تایی را متعادل می‌نامیم هرگاه همه برگ‌های آن در سطح h یا $h-1$ قرار داشته باشند.
- اگر $m=2$ باشد درخت را دودویی می‌نامیم.

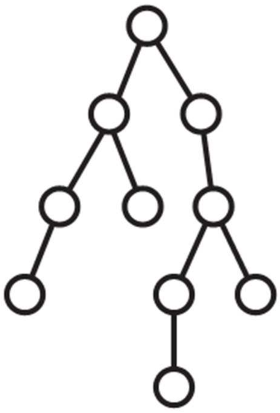
درخت ریشه‌دار

درخت دودویی متعادل





درخت دودویی تام



درخت دودویی

درخت ریشه‌دار

- **قضیه:** هر درخت m- تایی تام با i راس داخلی دارای mi+1 راس است.
- **نتیجه:** هر درخت دودویی با i راس داخلی دارای i+1 برگ است.

- **قضیه:** اگر T یک درخت دودویی با ارتفاع h و p راس باشد ، آنگاه

$$h + ۱ \leq p \leq ۲^{h+۱} - ۱$$

ایده اثبات: کران پایین برای مسیر جهتدار به طول p-1 برقرار است.

در هر درخت دودویی تعداد راس‌ها در هر سطح حداکثر دو برابر تعداد راس‌ها در سطح قبل است.

الگوریتم DFS

- ورودی: گراف G
- خروجی: جنگل فراگیر T (برای حفظ اطلاعات پدر و فرزندی در جنگل از متغیر pred استفاده می‌کنیم.)
- به هر راس u اندیس dfi(u) نسبت داده می‌شود

الگوریتم DFS

$O(\max\{n,m\})$

1- به ازای هر راس u قرار دهید $dfi(u)= 0$ و $pred(u) = 0$.

2- قرار دهید $k = 1$.

3- یک راس r با $dfi(r) = 0$ انتخاب کنید. قرار دهید $u = r$ ، $dfi(u) = k$ و $k = k+1$

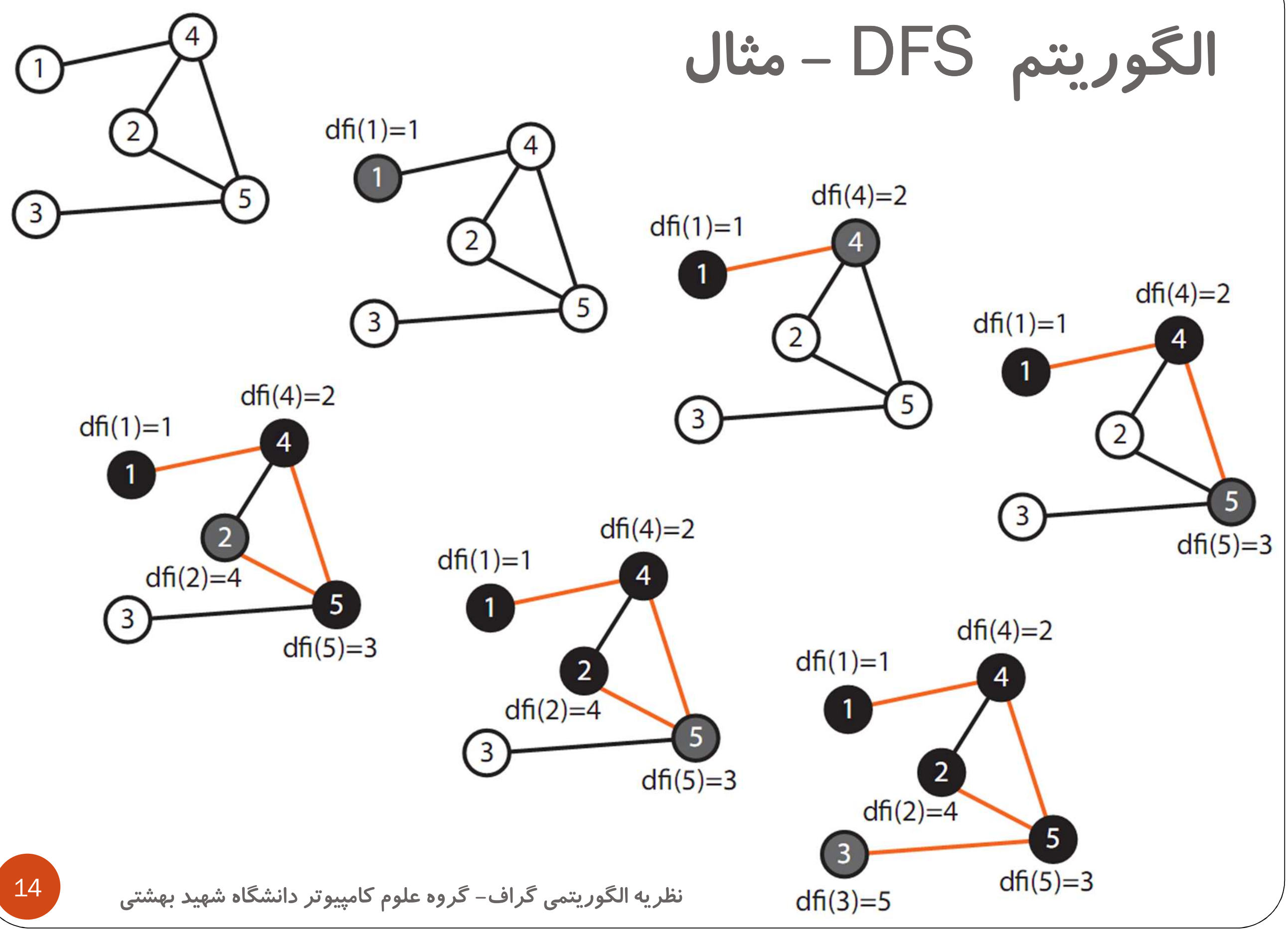
4- تا زمانی که $u \neq r$ مراحل زیر را تکرار کنید.

- اگر همه راس‌های مجاور u مشاهده شده‌اند ، قرار دهید $u= pred(u)$.
- در غیر این صورت ، فرض کنید v راس مجاور u و مشاهده نشده باشد ، قرار دهید

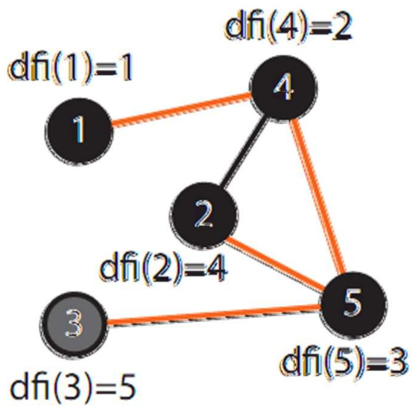
$pred(v) = u$ ، $dfi(v) = k$ و $u = v$ و $k = k+1$

5- اگر برای هر راس u داریم $dfi(u) \neq 0$ الگوریتم تمام شده. در غیر این صورت به مرحله 3 بروید.

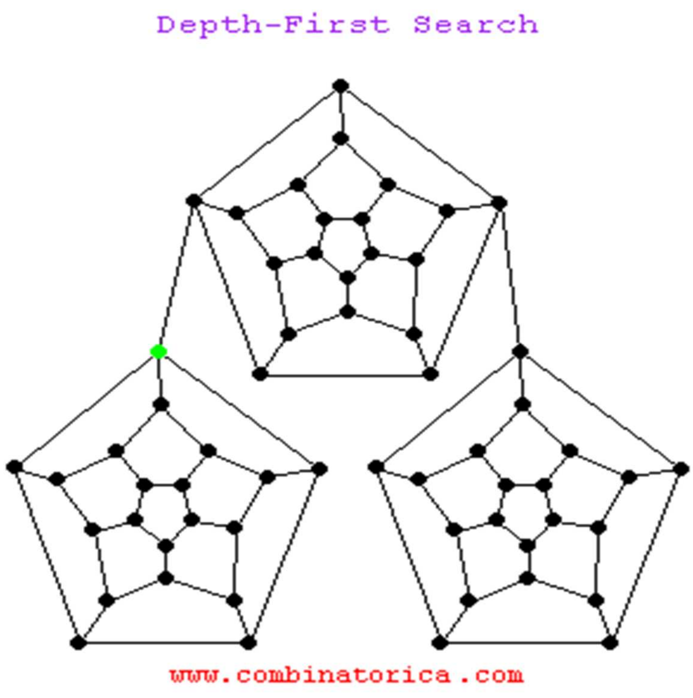
الگوریتم DFS – مثال



الگوریتم DFS – مثال



اجرای الگوریتم DFS



الگوریتم BFS

- ورودی: گراف G
- خروجی: یک جنگل فراگیر از گراف G
- برای هر راس یکی از وضعیت‌های مشاهده نشده $(status=1)$، درحال پردازش $(status=1)$ و پردازش شده $(status=3)$ وجود دارد.
- معمولا در پیاده سازی این الگوریتم از ساختمان داده صف استفاده می‌شود.

الگوریتم BFS

1- برای هر راس u در G قرار دهید $status(u) = 1$ و $k = 1$.

2- یک راس دلخواه u با $status(u)= 1$ انتخاب کنید ، قرار دهید $status(u)=2$ ، u را در صف وارد کنید و قرار دهید $pred(u) = 0$.

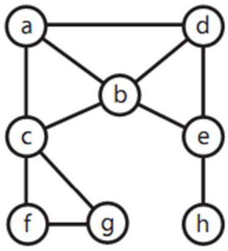
3- تا زمانی که صف خالی نشده ،

3-1- یک راس v از ابتدای صف بردارید. همه همسایگان v که وضعیت 1 دارند ، به وضعیت 2 ببرید و در صف وارد کنید.

3-2- قرار دهید $status(v) = 3$، $bfi(v) = k$ ، $pred(v) =u$ و $u=v$. و $k = k+1$.

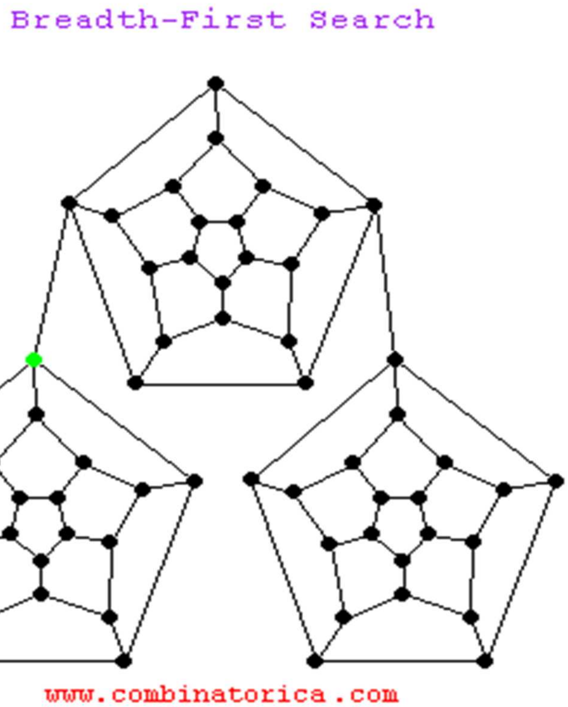
4- اگر راسی در گراف با وضعیت 1 باقی مانده است ، به مرحله 2 بروید. در غیر این صورت الگوریتم تمام شده است.

الگوریتم BFS – مثال



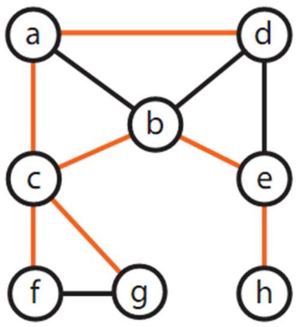
(1)

اجرای الگوریتم BFS



درخت فراگیر

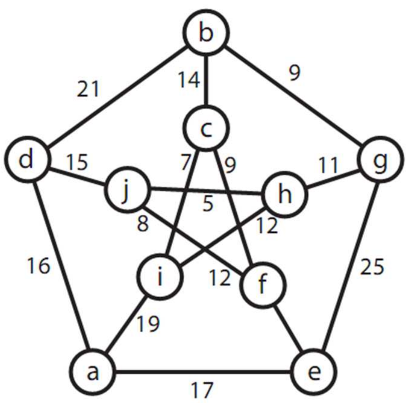
- زیرگرافي فراگیر از گراف G که درخت است ، درخت فراگیر نام دارد.

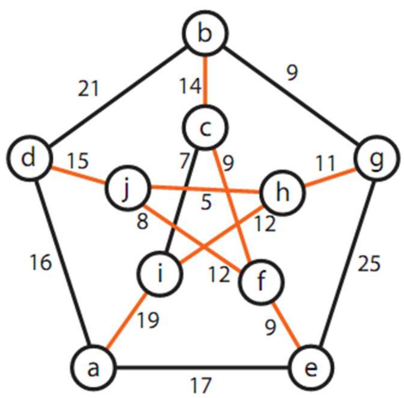


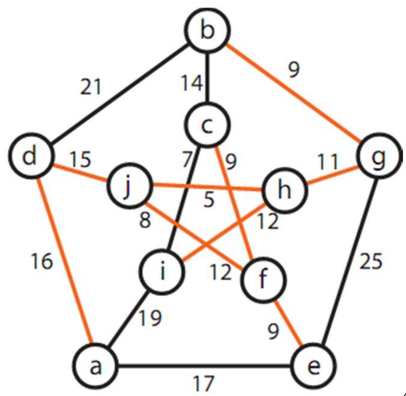
- آیا هر گرافي داراي درخت فراگیر است ؟
- آیا هر گراف همبندي داراي درخت فراگیر است ؟ چرا ؟
- آیا درخت فراگیر گراف یکتاست ؟
- چطور مي‌توان درخت فراگیر براي یك گراف ساخت ؟

درخت فراگیر

- در گراف وزن دار ، وزن یك زیرگراف برابر با مجموع وزن یال‌هاي آن است.







مساله کوچکترین درخت فراگیر

- **مساله کوچکترین درخت فراگیر (MST):** براي گراف وزن دار داده شده درخت فراگيري با كمترين وزن بيابيد.
- در یک گراف وزن‌دار درخت فراگیر لزوما یکتا نیست.
- در هر گراف که وزن یال‌ها متمایز است ، MST یکتاست.
- بهترین الگوریتم‌ها برای حل مساله الگوریتم‌های حریصانه هستند.
- کاربرد:
  - طراحي شبكه هاي خدمات شهري ،
  - بسياري از كاربردهاي تئوري ،

الگوریتم های تعیین کوچکترین درخت فراگیر

- در گراف‌های بدون وزن (یا وزن همه یال ها برابر باشد.)
  - الگوریتم BFS
  - الگوریتم DFS
- در گراف‌های وزن دار
  - الگوریتم کروسکال
  - الگوریتم پریم
  - الگوریتم بروفکا (در صورتی که وزن همه یال‌ها متمایز باشد.)

الگوریتم کروسکال

.1 [مقدار دهی اولیه S ، مجموعه یال‌های کوچکترین درخت فراگیر است.]

$$S \leftarrow \phi$$

.2 [مجموعه S افزایش می یابد.]

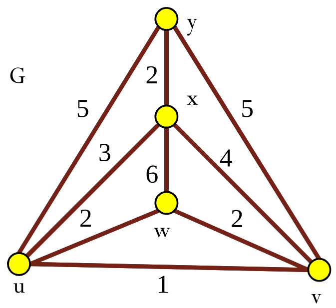
فرض کنید e یالی با کمترین وزن است ، $e \notin S$ و $S \cup \{e\}$ بدون دور است، قرار دهید

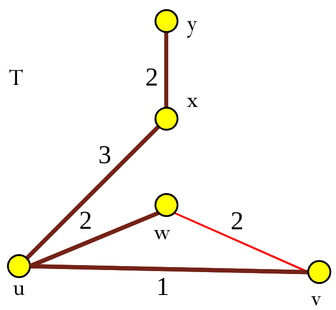
$$S \leftarrow S \cup \{e\}$$

.3 [این مرحله بررسی می‌کند کوچکترین درخت فراگیر ساخته شده.]

اگر $|S| = p - 1$ ، خروجی S است. در غیر این صورت به مرحله 2 بازگردید.

الگوریتم کروسکال- مثال





درستی الگوریتم کروسکال

- **قضیه:** در هر گراف وزن دار ، الگوریتم کروسکال کوچکترین درخت فراگیر را تولید مي كند.

ایده اثبات: فرض كنيد T خروجي الگوريتم و T' يك كوچكترين درخت فراگير باشد. اولين يال T كه در T' نيست را e_i بناميد.

$T'+e_i$ داراي دور است. يك يال e' كه در T نيست از اين دور برمي داريم.

$T'+e_i-e'$ يك درخت فراگير است.

اگر $w(e_i)<w(e')$ باشد ، $w(T)>w(T')$ مي شود كه امكان ندارد. اگر $w(e_i)>w(e')$ باشد ، e' در الگوريتم انتخاب مي شود. پس $w(e_i)=w(e')$ و $w(T)=w(T')$

الگوریتم پریم

1. [تعيين درخت اوليه]

u را یک راس دلخواه از G درنظر بگيريد و

2. [به روز کردن درخت T]

فرض کنید e یالی با کمترین وزن است که به راسی از T و راسی که در T نیست وصل است و قرار دهید

3. [تشخيص اين كه كوچكترين درخت فراگير ساخته شده يا نه.]

اگر $|E(T)| = p-1$ ، خروجی $E(T)$ است. در غیر این صورت به مرحله 2 بروید.

الگوریتم پریم – مثال