

طراحی الگوریتم

موسسه آموزش عالی پاسارگاد شیراز

الگوریتم حریصانه (greedy)

- در این روش می‌خواهیم بهترین گزینه را در یک لحظه انتخاب کنیم به طوری که در آن نه به گذشته نگاه کنیم و نه به آینده
- در واقع به انتخاب‌هایی که در گذشته انجام داده ایم و انتخاب‌هایی که در آینده انجام خواهیم داد توجهی نمی‌کنیم
- انتخاب بر اساس یک معیار خاص انجام می‌گیرد در واقع در روش حریصانه می‌خواهیم از بین حالت‌های مختلف که می‌توان به عنوان مجموعه جواب انتخاب کرد بهترین مجموعه را به دست آوریم

الگوریتم حریصانه (greedy)

- گام های مربوط به الگوریتم حریصانه:

- ۱- گام یا مرحله انتخاب:

- در این مرحله یک عنصر را به مجموعه جواب اضافه میکنیم که انتخاب این عنصر بر اساس یک معیار خاص است.

- ۲- تحقیق عملی بودن:

- در این قسمت بررسی می شود که آیا عنصری که انتخاب کرده ایم ما را به جواب می رساند؟

- ۳- تحقیق درباره اینکه جواب نهایی حاصل شده است یا خیر

- اگر به جواب نهایی رسیده باشیم کار به پایان می رسد و اگر به جواب نهایی نرسیده باشیم به مرحله انتخاب باز می گردیم.

الگوریتم حریصانه (greedy)

• یادآوری

- گراف: از یک سری گره و یال تشکیل شده است که مجموعه گره را با V و مجموعه یال را با E نشان می دهیم. $G(V, E)$
- گراف ها میتوانند جهت دار و یا بدون جهت باشند اگر گراف جهت دار باشد آنگاه یال های $\langle V1, V2 \rangle$ و $\langle V2, V1 \rangle$ با هم برابر نیستند ولی اگر بدون جهت باشند آنگاه یال های $(V1, V2)$ و $(V2, V1)$ با هم برابر هستند.
- گراف وزن دار: گرافی که هر کدام از یال های آن دارای وزن باشد.
- گراف همبند: گرافی که بین تمام گره های آن حداقل یک مسیر وجود داشته باشد (گراف همبند با حداقل یال درخت)

الگوریتم حریصانه (greedy)

• یادآوری

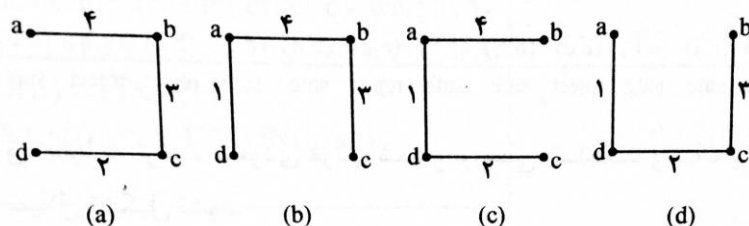
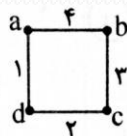
• درخت : گراف همبندی که دور نداشته باشد. تعداد یال های آن یکی کمتر از تعداد گره ها است یعنی اگر تعداد گره های درخت n باشد تعداد یال های آن $n-1$ یال می باشد.

• درخت پوشا (spanning tree)

• از گراف همبند به وجود می آید. تعداد گره های آن با تعداد گره های گراف

برابر است. درخت پوشا لزوماً یکتا نیست.

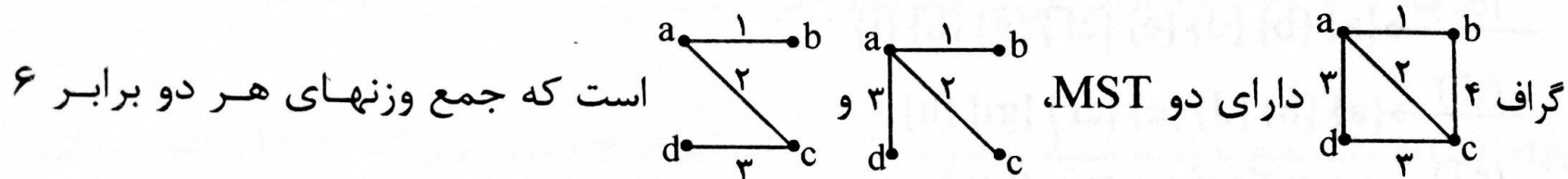
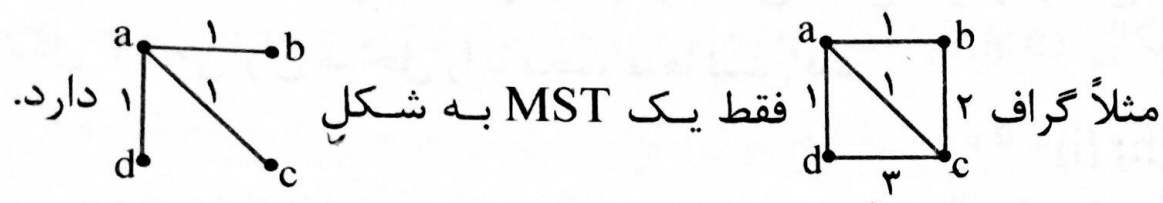
مثلاً گراف G : دارای ۴ درخت پوشاست که عبارتند از:



الگوریتم حریصانه (greedy)

یادآوری

- درخت پوشای مینیمم (Minimum Spanning Tree : MST)
- درخت پوشایی است که از یک گراف وزن دار همبند به وجود می‌آید و مجموعه یال های این درخت مینیمم است.
- درخت پوشا مینیمم لزوماً یکتا نیست.



است.

الگوریتم حریصانه (greedy)

- الگوریتم های پیدا کردن درخت پوشای مینیمم

- ۱- پریم Prim

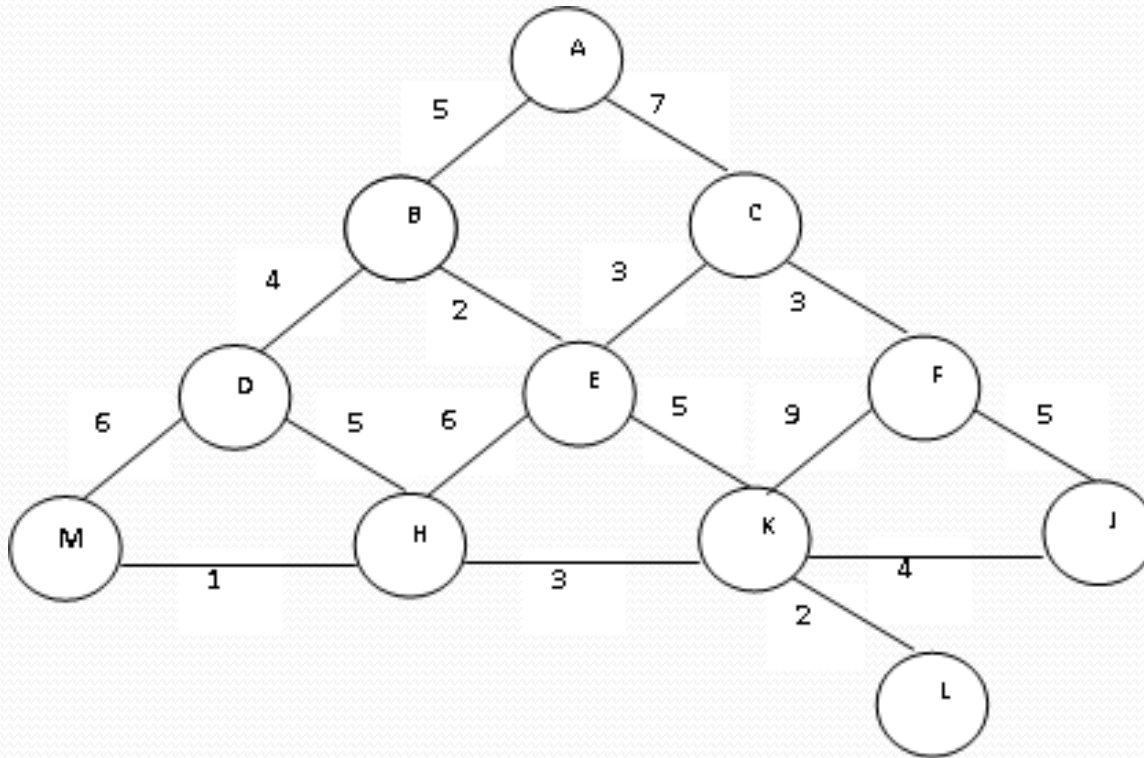
- در این الگوریتم فرض می کنیم که از گره ای مثل $V1$ شروع به ساخت درخت کنیم به طوری که مجموع یال های انتخاب شده را با F و مقدار اولیه آن تهی است. گره های انتخاب شده را با Y نشان می دهیم که مقدار اولیه آن $V1$ می باشد. در این الگوریتم از بین یال های انتخاب نشده یال با کمترین هزینه را با شرایطی انتخاب می کنیم که یک سمت یال در مجموعه مربوط به Y وجود داشته باشد و سمت دیگر آن در Y وجود نداشته باشد. سپس یال اضافه شده را به مجموعه F اضافه می کنیم و آن سمتی از یال را (گره ای از یال) که در مجموعه Y وجود نداشته به عنوان گره جدید به مجموعه Y اضافه می کنیم و عملیات را تا زمانی ادامه می دهیم که تمام گره ها به مجموعه Y اضافه شوند یا تعداد یال های انتخاب شده یکی کمتر از تعداد کل گره ها باشد.

- مرتبه اجرایی الگوریتم پریم

- چون باید n گره انتخاب شود و پس از انتخاب هر گره باید تمام یال های آن چک شود بنابراین مرتبه اجرایی $O(n^2)$ خواهد بود.

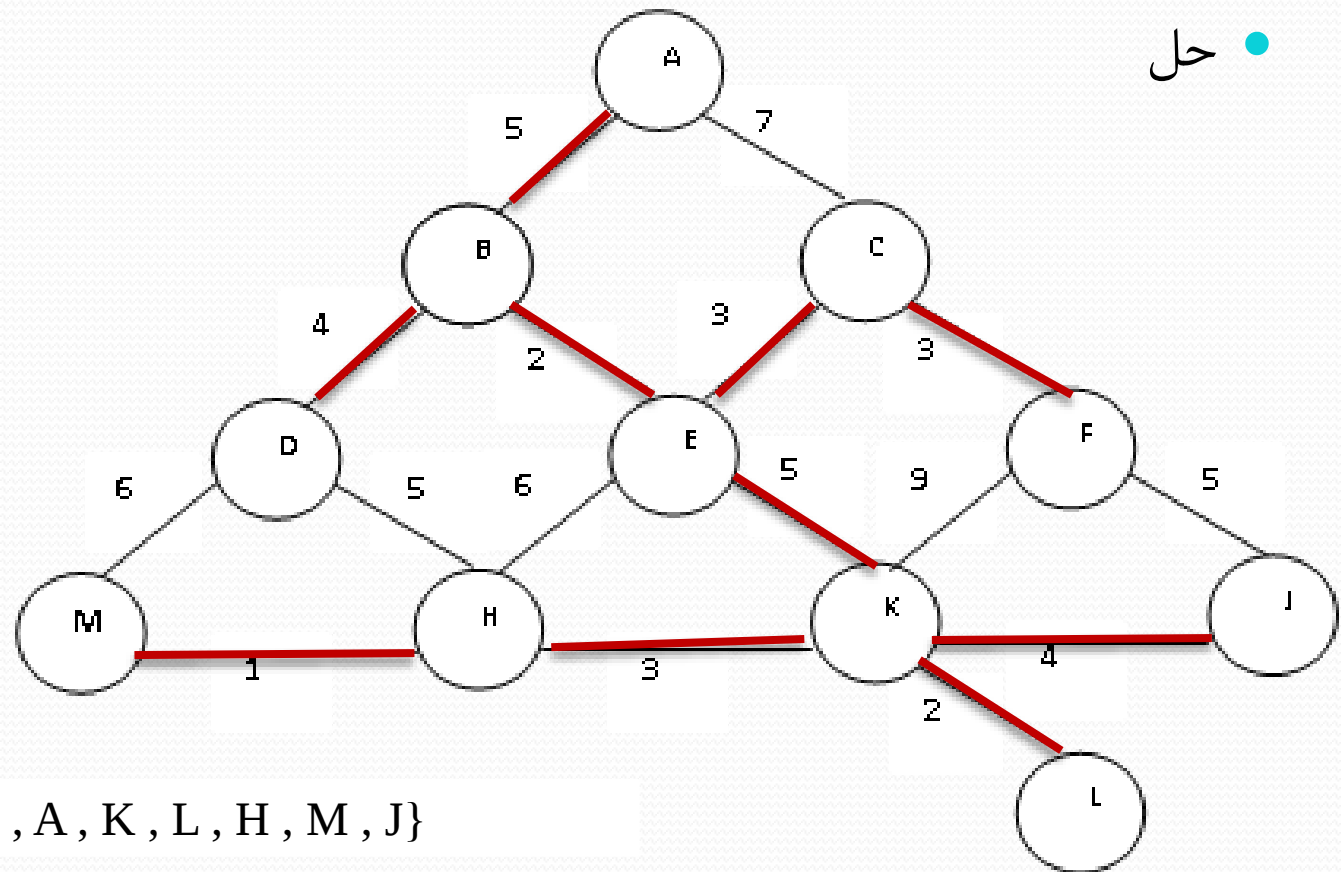
الگوریتم حریصانه (greedy)

- مثال به روش پریم عملیات ساخت درخت پوشای مینیمم را برای گراف روبرو با شروع از گره E انجام دهید.



الگوریتم حریصانه (greedy)

• حل



$Y = \{E, B, C, F, D, A, K, L, H, M, J\}$

$F = \{(E, B), (E, C), (C, F), (B, D), (B, A), (E, K), (K, L), (K, H), (H, M), (K, J)\} = 32$

الگوریتم حریصانه (greedy)

- الگوریتم های پیدا کردن درخت پوشای مینیمم

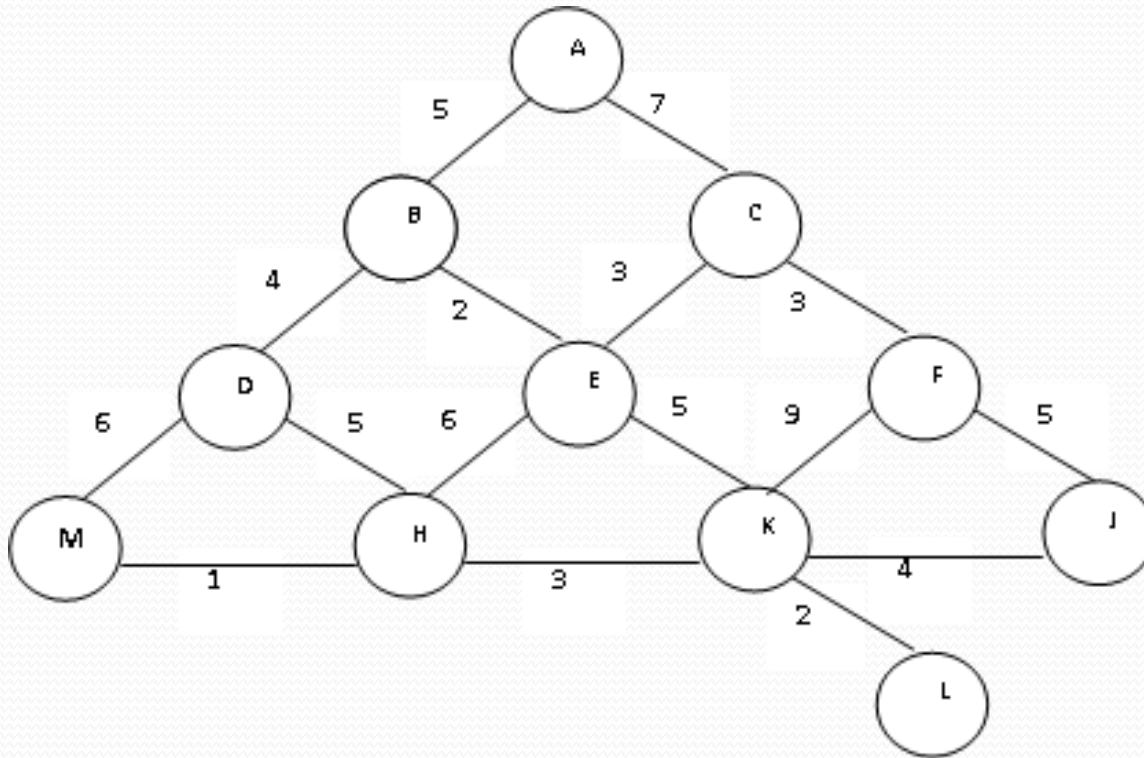
- ۲- الگوریتم کراسکال (Kruskal)

- در این الگوریتم ابتدا تمام یالها را به ترتیب صعودی مرتب می کنیم سپس در مرحله انتخاب یال با کمترین وزن را انتخاب می کنیم و در مرحله تحقیق عملی بودن چک میکنیم که این انتخاب باعث ایجاد دور در درخت می شود یا خیر اگر باعث ایجاد دور شد به سراغ یال بعدی با کمترین هزینه می رویم ولی اگر ایجاد دور نکرد یال را به مجموعه جواب اضافه می کنیم و در نهایت عملیات انتخاب کردن را تا انتخاب $n-1$ یال به مجموعه جواب ادامه می دهیم

- مرتبه اجرایی : مهمترین کار در این الگوریتم مرتب کردن یال هاست که اگر تعداد یال ها e تا باشد مرتبه اجرایی الگوریتم $O(e \log e)$ می شود.

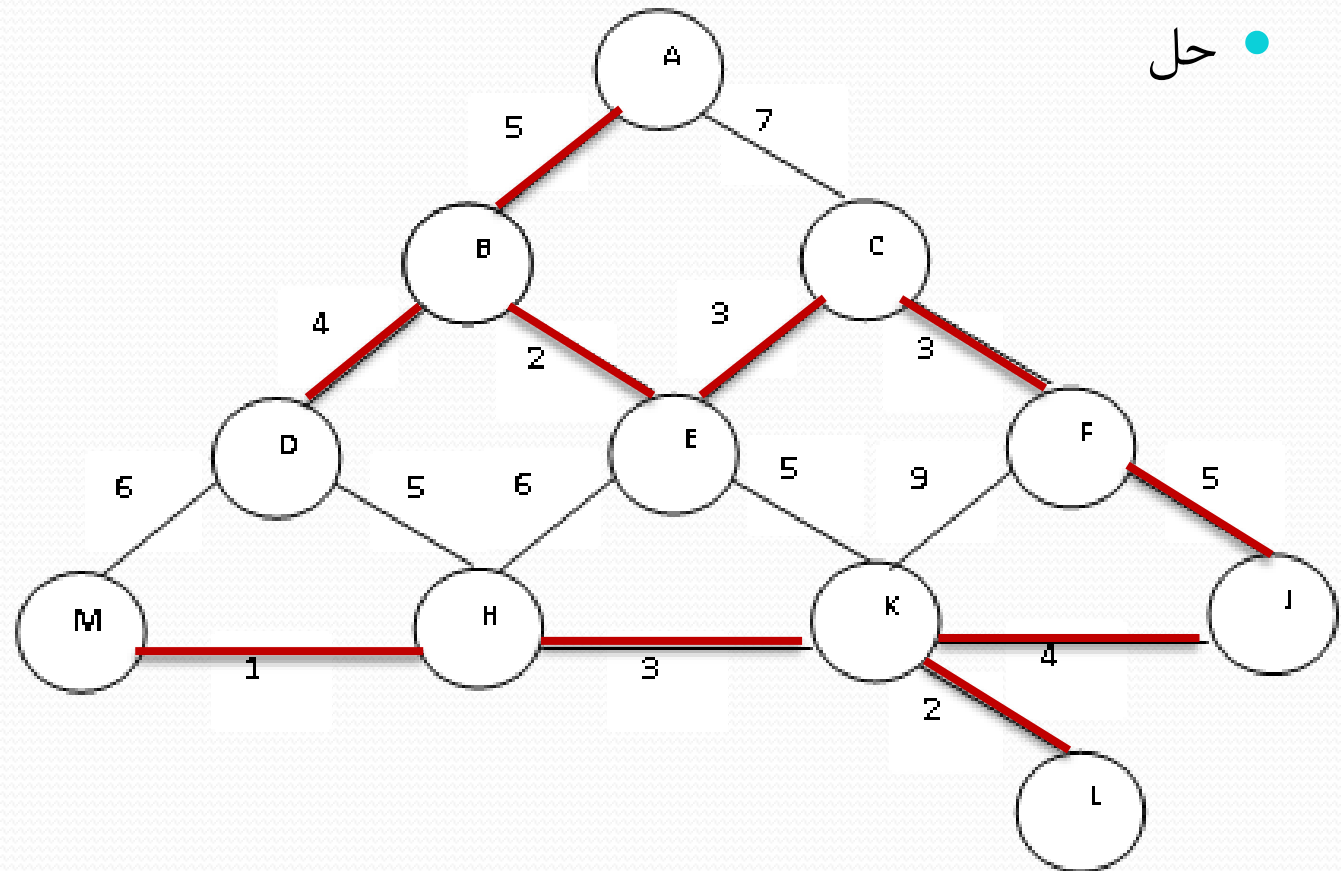
الگوریتم حریصانه (greedy)

- مثال به روش کراسکال عملیات ساخت درخت پوشای مینیمم را برای گراف زیر انجام دهید.



الگوریتم حریصانه (greedy)

• حل



$$F = \{(M,H), (K,L), (B,E), (C,F), (K,H), (C,E), (K,J), (B,D), (F,J), (A,B)\} = 32$$

الگوریتم حریصانه (greedy)

- نکته: در صورتیکه تعداد یال ها کم باشد از الگوریتم کراسکال برای به دست آوردن درخت پوشا استفاده می کنیم و در صورتیکه تعداد یال ها زیاد باشد از الگوریتم پریم استفاده می کنیم. کراسکال به تعداد یال وابسته است ولی پریم به تعداد گره وابسته است.

الگوریتم حریصانه (greedy)

- الگوریتم هافمن

- در این الگوریتم می‌خواهیم برای هر کاراکتر کد ایجاد کنیم به طوری که کاراکترها با تعداد تکرار بیشتر دارای طول کد کمتری باشند و کاراکترها با تعداد تکرار کمتر دارای طول کد بیشتری باشند. این الگوریتم به صورت زیر انجام می‌شود

- ۱- ابتدا باید درخت هافمن را ساخت

- در آغاز هر کاراکتر و تعداد تکرار آن را به عنوان یک گره در نظر می‌گیریم سپس از بین گره‌ها دو گره که کمترین مقدار را دارند انتخاب می‌کنیم و آن دو را از مجموعه گره‌ها حذف می‌کنیم و سپس از این دو گره، گره حاصل جمع را ایجاد می‌کنیم و گره حاصل شده را به مجموعه گره‌ها اضافه می‌کنیم و دوباره عملیات انتخاب و ادغام را انجام می‌دهیم. این عملیات را تا زمانی ادامه می‌دهیم که تنها یک گره باقی بماند در این حالت درخت هافمن ایجاد شده است

الگوریتم حریصانه (greedy)

- الگوریتم هافمن

- ۲- یال های درخت را به این صورت مقداردهی میکنیم که هر گره که دو یال دارد به یال سمت راست آن مقدار ۱ و به یال سمت چپ آن مقدار صفر اختصاص داده می شود .

- ۳- پیدا کردن کد هر کاراکتر: برای به دست آوردن کد هر کاراکتر از ریشه شروع می کنیم و مقدار یال ها تا رسیدن به کاراکتر مورد نظر را یادداشت می کنیم که همان کد کاراکتر است.

الگوریتم حریصانه (greedy)

● مثال

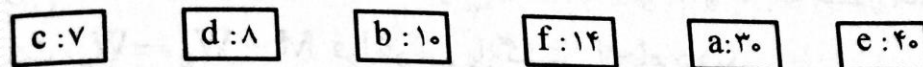
فرض کنید متنی شامل حروف a و b و c و d و e و f می باشد و فراوانی این حروف در جدول زیر آمده است، می خواهیم کدی بهینه برای این حروف تولید کنیم.

کاراکتر	a	b	c	d	e	f
فراوانی	۳۰	۱۰	۷	۸	۴۰	۱۴

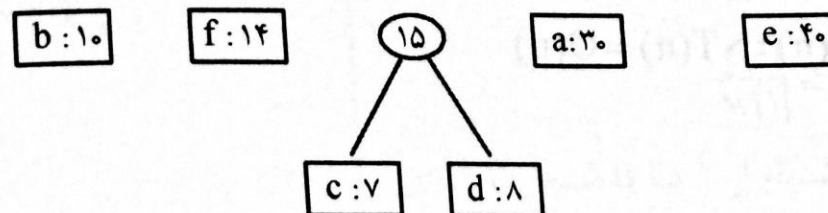
پاسخ: ابتدا دو کاراکتر با کمترین فراوانی انتخاب می شوند و با آنها یک درخت ساخته می شود و جمع فراوانی آنها در ریشه درخت نوشته می شود و آن دو فراوانی از جدول حذف می شوند و جمع فراوانی آنها به جدول اضافه می شود. سپس دو فراوانی مینیمم دیگر انتخاب می شوند و به همین ترتیب. مراحل اجرا در ادامه آمده است.

الگوریتم حریصانه (greedy)

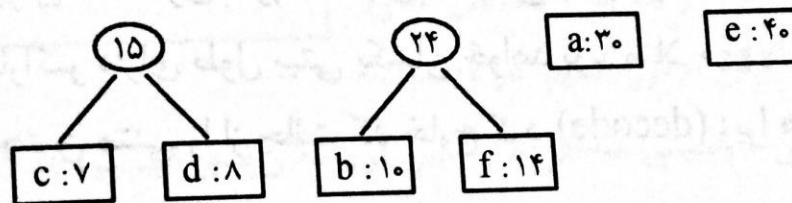
الف) فراوانی‌ها را به ترتیب صعودی مرتب می‌کنیم:



ب) **c** و **d** انتخاب می‌شوند و مجموع فراوانی آنها یعنی ۱۵ بعد از ۱۴ قرار می‌گیرد:

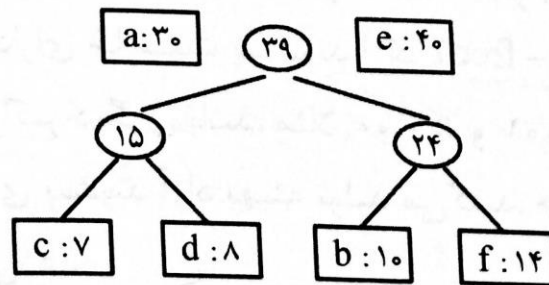


ج) حال **b** و **f** انتخاب می‌شوند و با آنها درخت ساخته می‌شود:

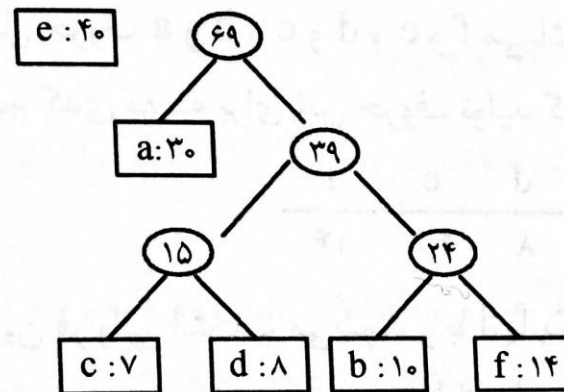


الگوریتم حریصانه (greedy)

(د) با ۱۵ و ۲۴ درخت ساخته می‌شود:

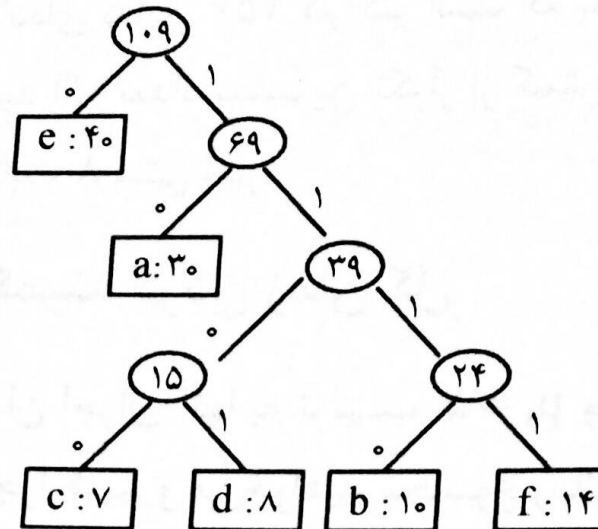


(ه) با ۳۰ و ۳۹ درخت ساخته می‌شود:



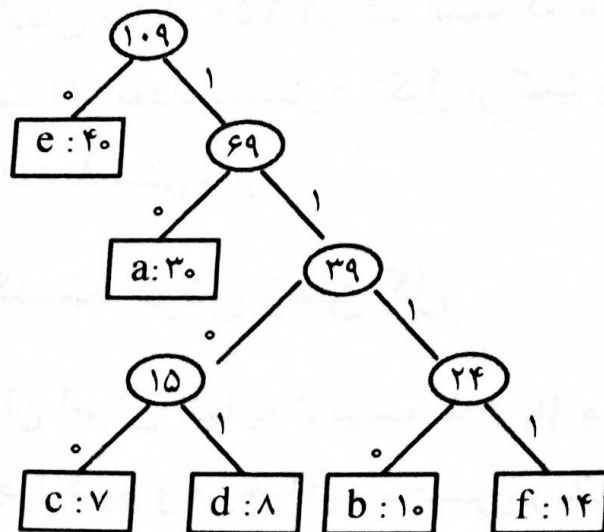
الگوریتم حریصانه (greedy)

و) در نهایت با ۴۰ و ۶۹ درخت می‌سازیم و بعد از ساخت درخت، روی یال‌های چپ و روی یال‌های راست ۱ می‌نویسیم. بدین ترتیب کد هر کاراکتر بدست می‌آید:



الگوریتم حریصانه (greedy)

و) در نهایت با ۴۰ و ۶۹ درخت می‌سازیم و بعد از ساخت درخت، روی یال‌های چپ و روی یال‌های راست ۱ می‌نویسیم. بدین ترتیب کد هر کاراکتر بدست می‌آید:



$e=0$, $a=10$, $c=1100$, $d=1101$, $b=1110$, $f=1111$

روش عقبگرد (Backtrack)

- فرض کنید می خواهیم در غار هزارتوی حرکت کنیم و هنگامی که به بن بست برخورد کنیم باید از همان راهی که آمده ایم برگردیم و بر سر چند راهی ها مسیر خود را عوض کنید حال اگر بر سر برخی از مسیرها تابلوی وجود داشته باشد که به شما بگوید انتها این مسیر بن بست است آن گاه صرفه جویی زیادی در وقت خود خواهید کرد این تکنیک همان الگوریتم عقبگرد است

روش عقبگرد (Backtrack)

- روش عقبگرد برای حل مسائلی به کار میرود که در آن دنبال ای از اشیا از یک مجموعه مشخص انتخاب می شود به گونه ای که این دنباله ملاک و قانونی خاص را دنبال کند. مشهور ترین مسئله کلاسیک برای این تکنیک مسئله n وزیر است.
- در این مسئله می خواهیم n وزیر را در یک صفحه شطرنج $n \times n$ به گونه ای قرار دهیم که هیچ وزیری نتواند دیگری را تهدید کند.

روش عقبگرد (Backtrack)

- روش عقبگرد برای حل مسائلی به کار میرود که در آن دنبال ای از اشیا از یک مجموعه مشخص انتخاب می شود به گونه ای که این دنباله ملاک و قانونی خاص را دنبال کند. مشهور ترین مسئله کلاسیک برای این تکنیک مسئله n وزیر است.
- در این مسئله می خواهیم n وزیر را در یک صفحه شطرنج $n \times n$ به گونه ای قرار دهیم که هیچ وزیری نتواند دیگری را تهدید کند.

روش عقبگرد (Backtrack)

مساله n وزیر

می خواهیم n وزیر را در صفحه شطرنجی $n \times n$ طوری قرار دهیم که هیچ دو وزیری یکدیگر را تهدید نکنند (سطر، ستون و قطر آنها یکسان نباشد).

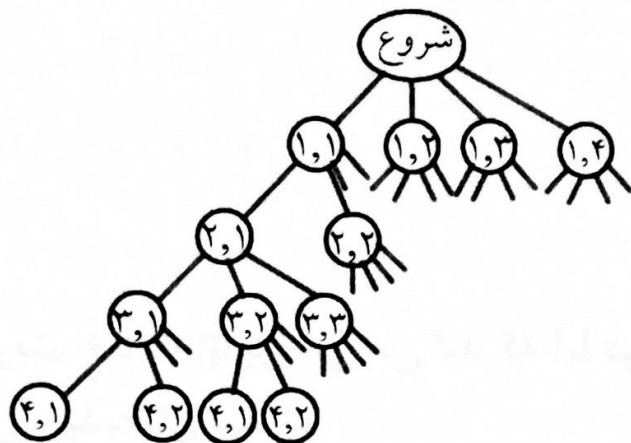
به عنوان مثال به ازای $n = 8$ یک جواب در شکل زیر آمده است:

	۱	۲	۳	۴	۵	۶	۷	۸
۱				*				
۲						*		
۳								*
۴		*						
۵							*	
۶	*							
۷			*					
۸					*			

روش عقبگرد (Backtrack)

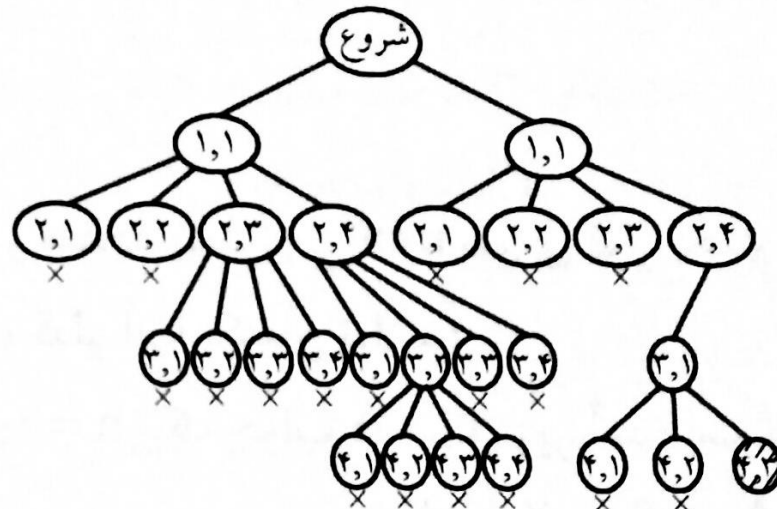
فرض کنید به ازای $n = 4$ می‌خواهیم مسئله را حل کنیم. با توجه به اینکه هیچ دو وزیری نمی‌توانند در یک سطر قرار بگیرند، برای هر وزیر ۴ حالت وجود دارد بنابراین تعداد کل حالات ممکن $4 \times 4 \times 4 \times 4 = 256$ می‌باشد. این ۲۵۶ حالت را می‌توان با درخت فضای حالت نشان داد.

در درخت، گره‌ها با زوج (i, j) نشان داده می‌شوند که i وزیر i ام است که در سطر i ام قرار دارد و j شماره ستونی است که وزیر در آن قرار دارد. مثلاً زوج $(1, 2)$ یعنی وزیر سطر دوم در ستون یک قرار دارد:



روش عقبگرد (Backtrack)

اولین مسیر در درخت حالت فوق $\{(1, 1), (2, 1), (3, 1), (4, 1)\}$ می باشد مسیر دوم در درخت $\{(2, 1), (3, 1), (4, 1), (1, 1)\}$ و ... می باشد. بررسی همه مسیرها معقول نیست و می توان با استفاده از علائمی بررسی را کاهش داد و درخت را هرس کرد. مثلاً هیچ دو وزیری نباید در یک ستون یا هم قطر باشند. شکل زیر بخشی از درخت هرس شده را نشان می دهد. گره های که امیدبخش نیست با $\times$ مشخص شده.



روش عقبگرد (Backtrack)

اولین مسیر در درخت حالت فوق $\{(1 \text{ و } 1) (2 \text{ و } 1) (3 \text{ و } 1) (4 \text{ و } 1)\}$ می باشد مسیر دوم در درخت $\{(2 \text{ و } 1) (3 \text{ و } 1) (4 \text{ و } 1) (1 \text{ و } 1)\}$ و ... می باشد. بررسی همه مسیرها معقول نیست و می توان با استفاده از علائمی بررسی را کاهش داد و درخت را هرس کرد. مثلاً هیچ دو وزیری نباید در یک ستون یا هم قطر باشند. شکل زیر بخشی از درخت هرس شده را نشان می دهد. گره ای که امیدبخش نیست با $\times$ مشخص شده.

1
2
3
4

	1	2	3	4
1		*		
2				*
3	*			
4			*	

