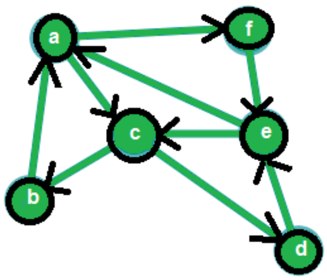
درس طراحی الگوریتم دانشکده ریاضی. دانشگاه صنعتی خواجه نصیرالدین طوسی کتاب مرجع: طراحی الگوریتم، جان کلاینبرگ، اوا تاردش	بهار ۹۹

چند مسئله مربوط به گرافهای جهت دار

۱ ذخیره‌سازی گرافهای جهت دار

گراف جهت دار، مانند گراف غیر جهت دار، به دو طریق ماتریس مجاورتی و لیست مجاورتی قابل نمایش و ذخیره‌سازی است.
در ماتریس مجاورتی اگر یال (i,j) وجود داشته باشد آنگاه درایه $A_{i,j}$ ماتریس برابر با 1 خواهد بود و در غیر اینصورت برابر با 0 است. یک مثال در شکل زیر نمایش داده شده است.



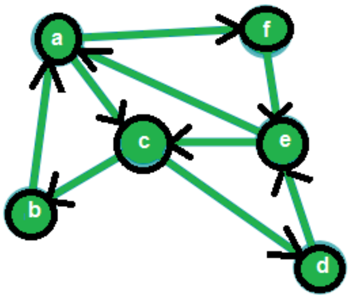
$$A = \begin{array}{c} \\ a \\ b \\ c \\ d \\ e \\ f \end{array} \begin{array}{c} \begin{array}{cccccc} a & b & c & d & e & f \end{array} \\ \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \end{array}$$

در لیست مجاورتی، برای هر راس $u \in V$ دو لیست نگه‌داری می‌شود. لیست ورودی $In(u)$ که مجموعه رئوسی است که از آنها به u یالی وجود دارد. لیست خروجی $Out(u)$ که مجموعه رئوسی است که از u به آنها یالی وجود دارد. برای نمایش لیست مجاورتی گراف بالا بصورت زیر است.

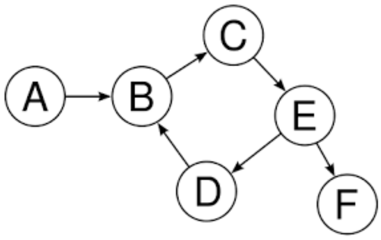
a	IN	b, e
	OUT	c, f
b	IN	c
	OUT	a
c	IN	a, e
	OUT	b, d
d	IN	c
	OUT	e
e	IN	f, d
	OUT	a, c
f	IN	a
	OUT	e

۱.۱ تشخیص گراف قویاً همبند

گراف جهت دار G را قویاً همبند گویند اگر و فقط اگر از هر راس G مسیری به همه رئوس گراف وجود داشته باشد. به عبارت دیگر، در گراف قویاً همبند بین هر دو راس مسیری وجود دارد. برای مثال گراف زیر قویاً همبند است.



اما گراف زیر قویاً همبند نیست (چون از بقیه رئوس مسیری به A وجود ندارد.)



۲.۱ الگوریتم تشخیص قویاً همبند بودن گراف

برای تشخیص اینکه آیا گراف جهت دار $G = (V, E)$ قویاً همبند است یا نه، یک راه بدیهی این است از هر راس $u \in V$ یک پیمایش $BFS(u)$ انجام دهیم (از DFS هم می‌توان استفاده کرد.) اگر در یکی از این پیمایشها همه رئوس V ملاقات نشدند، آنگاه G قویاً همبند نیست و در غیر اینصورت قویاً همبند است. این الگوریتم از خود تعریف قویاً همبند بودن استفاده می‌کند و روشن است که درست است اما هزینه اجرایش بالاست. هر اجرای BFS به زمان $O(m + n)$ نیاز دارد. لذا در مجموع زمان اجرا $O(n(m + n))$ خواهد بود.

یک ایده بهتر، استفاده از متعدی بودن رابطه دسترسی است. اگر مسیری از u به v داشته باشیم و مسیری از v به w هم داشته باشیم آنگاه بدیهی است مسیری u به w وجود خواهد داشت.

با توجه به مشاهده بالا، این ایده را پیشنهاد می‌دهیم: فرض کنید x یک راس دلخواه گراف G باشد. اگر از x همه رئوس گراف قابل دسترسی باشند و همچنین از همه رئوس گراف مسیری به x وجود داشته باشد آنگاه G قویاً همبند خواهد بود. دلیل این مطلب را می‌توان بصورت زیر بیان کرد. فرض کنید راس x چنین خاصیتی داشته باشد. حال دو راس دلخواه y و z را در نظر بگیرید. چون از y به x مسیری وجود دارد و از x هم به z مسیری وجود دارد،

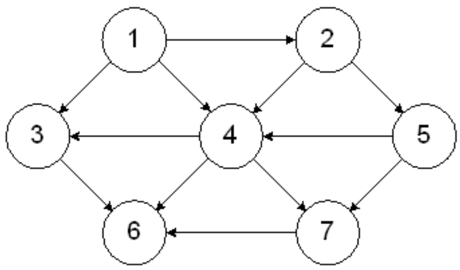
بنا به متعدی بودن رابطه دسترسی، پس مسیری از y به z خواهیم داشت. به همین منوال می‌توان مشاهده کرد که مسیری از z به y وجود دارد. پس گراف قویاً همبند است.

ایده بالا با اجرای تنها دو پیمایش BFS قابل پیاده‌سازی است. اگر بخواهیم وجود مسیر از x به همه رئوس را چک کنیم، می‌توانیم از $BFS(x)$ استفاده کنیم. اما اگر بخواهیم وجود مسیر از همه رئوس به راس x را با اجرای یک BFS چک کنیم، نیاز به یک ترفند داریم. جهت همه یالهای G را برعکس می‌کنیم و سپس در گراف ایجاد شده G' الگوریتم $BFS(x)$ را اجرا می‌کنیم. اگر همه رئوس ملاقات شدند، یعنی اینکه هر راسی مسیری به x در گراف G وجود دارد.

لازم به ذکر است که عوض کردن جهت یالها در زمان $O(n)$ قابل انجام است (چرا؟). لذا ایده جدید در زمان $O(m+n)$ قابل اجرا است.

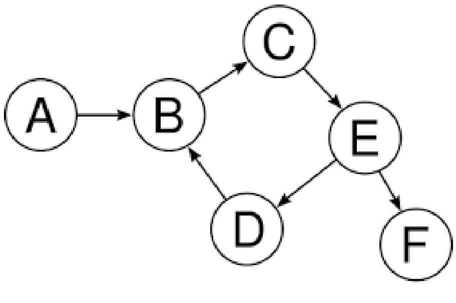
۲ ترتیب توپولوژیکی در گرافهای جهت دار بدون دور

تعریف: گراف G را یک گراف جهت دار بدون دور، اصطلاحا یک DAG (Directed Acyclic Graph) گویند اگر G جهت دار باشد و هیچ دوری نداشته باشد. برای مثال گراف زیر یک DAG است.



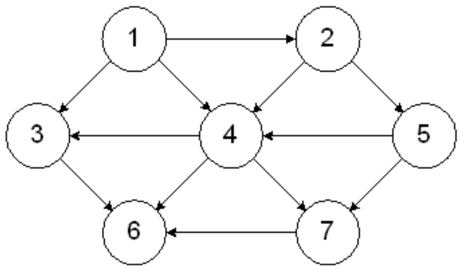
توجه کنید وجود دور به این معنی است که برای مثال دو راس x و y نداشته باشیم که از x به y یک مسیر از y به x مسیری وجود داشته باشد.

گراف زیر DAG نیست چون دور BCED وجود دارد.



تعریف: ترتیب $v_1, v_2, \ldots, v_n$ از رئوس G را یک ترتیب توپولوژیکی گویند اگر هیچ دو راس v_i و v_j در ترتیب داده شده نباشد که $i < j$ و یال (v_j, v_i) در G موجود باشد. به عبارت دیگر، در ترتیب داده شده، یال به سمت عقب وجود نداشته باشد.

برای مثال ترتیب $1, 2, 5, 4, 7, 3, 6$ یک ترتیب توپولوژیکی برای گراف شکل زیر است. شایان ذکر است که ترتیب توپولوژیکی رئوس یک گراف منحصر بفرد نیست و می‌تواند متعدد باشد. ترتیب $1, 2, 5, 4, 3, 7, 6$ یک ترتیب توپولوژیکی دیگر برای گراف زیر است.



چه زمانی یک گراف جهت دار ترتیب توپولوژیکی دارد؟ روشن است، وجود دور (جهت دار) ناقض وجود ترتیب توپولوژیکی خواهد شد. چون رئوس دور را به هر ترتیبی بنویسیم یک یال عقبگرد ظاهر می‌شود.

پس آیا می‌توان گفت هر گراف جهت دار بدون دور، به عبارت دیگر هر DAG یک ترتیب توپولوژیکی دارد؟ جواب این سوال مثبت است. در واقع الگوریتمی وجود دارد که با داشتن یک DAG یک ترتیب توپولوژیکی برای رئوس آن تولید می‌کند. قبل از توصیف این الگوریتم، لم زیر را بیان می‌کنیم که در واقع ایده اصلی الگوریتم است.

لم : در هر گراف جهت دار بدون دور، یک راس وجود دارد که یال ورودی ندارد.

اثبات این لم بر عهده دانشجو.

الگوریتم تولید ترتیب توپولوژیکی برای گرافهای جهتدار بدون دور، بصورت زیر عمل می‌کند:

۱. G گراف ورودی است.
۲. لیست T در ابتدا تهی است.
۳. تا زمانی که مجموعه رئوس G تهی نباشد، کارهای زیر را انجام بده.
   - در گراف G راس x پیدا کن که یال ورودی نداشته باشد.
   - راس x را از گراف G حذف کن و آن را به انتهای T اضافه کن.
۴. لیست T را به عنوان یک ترتیب توپولوژیکی از رئوس G گزارش کن.

در الگوریتم بالا به چگونگی حذف راس از گراف اشاره‌ای نشده است. پیاده‌سازی الگوریتم بالا در زمان $O(n^2)$ برای دانشجو باید ساده باشد (حذف راس از گراف جهت دار در چه زمانی قابل پیاده‌سازی است؟). البته یک پیاده‌سازی وجود دارد که در زمان $O(n+m)$ قابل اجراست. برای جزئیات بیشتر کتاب مرجع را ببینید.