کوییز شماره پنج
گراف

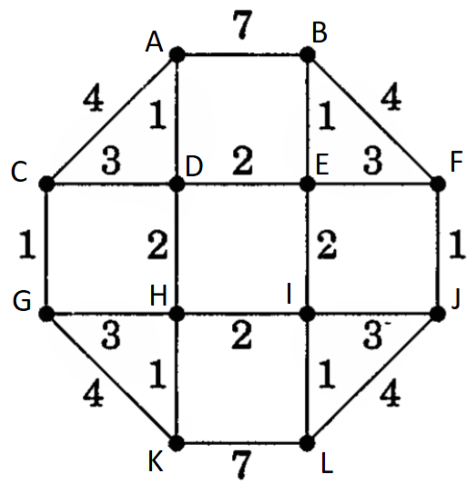
دانشکده مهندسی برق و کامپیوتر
ساختمان های داده و الگوریتم - پاییز 1400

استاد: دکتر هشام فیلی
طراح : ملیکا حیدری دستجردی
بارم 100 نمره

---

1) گراف زیر را در نظر بگیرید. طبق جدول زیر راس متناظر با باقیمانده جمع دو رقم سمت راست شماره دانشجویی خود بر 5 را تعیین کرده و با شروع از آن دو الگوریتم dfs و bfs را اجرا کنید. پیمایش طبق الگوریتم bfs تا عمق 3 کافی است. لطفا مراحل اجرای الگوریتم را روی گراف مشخص کنید و در انتها فاصله هر راس از مبدا را نشان دهید. ترتیب فرزندان یک گره به ترتیب حروف الفبا باشد.

راس شروع متناظر	باقیمانده جمع دو رقم سمت راست شماره دانشجویی بر 5
A	0
B	1
C	2
D	3
E	4



پاسخ:

وضعیت صف رئوس و فاصله هر راس به ازای اجرای BFS از راس A:

A_0

$B_7,\ C_4,\ D_1$

$C_4,\ D_1,\ E_8,\ F_{11}$

$D_1,\ E_8,\ F_{11},\ G_5$

$E_8,\ F_{11},\ G_5,\ H_3$

$F_{11},\ G_5,\ H_3,\ I_{10}$

$G_5,\ I_{10},H_3,\ J_{12}$

$I_{10},H_3,\ J_{12}\ ,\ K_9$

$H_3,\ J_{12}\ ,\ K_9$

$J_{12}\ ,\ K_9$

K_9

مقادیر starting time به ازای اجرای DFS از راس A

A	B	C	D	E	F	G	H	I	J	K	L
1	2	9	8	3	4	10	7	6	5	11	12

فواصل هر راس به ازای اجرای DFS از راس A َ

A	B	C	D	E	F	G	H	I	J	K	L
0	7	22	19	8	11	23	17	15	12	27	34

2) یک گراف n راسی با رئوس 1 تا n در نظر بگیرید. برای نشان دادن این گراف راه های زیادی وجود دارد. یکی از این راه ها ساختن یک آرایه $p_1, p_2, \dots p_n$ است که p_i parent راس iام را مشخص میکند. parent ریشه هم خودش است. بدین ترتیب یک آرایه میتواند آرایه نشانگر یک درخت باشد اگر

- دقیقا یک اندیس r وجود داشته باشد که $p_r = r$ و آن راس ریشه درخت است.
- برای هر یک از n-1 راس دیگر، یالی از راس i به p_i وجود داشته باشد.

یک دنباله معتبر است اگر طبق تعریف فوق یک درخت ریشه دار تولید کند. برای مثال به ازای n=3 ، دنباله های (2,1,3 و (1,2,2) ,(2,3,1) معتبر نیستند. یک دنباله $a_1, a_2, \dots a_n$ به شما داده شده است که لزوما معتبر نیست. با اعمال کمترین تغییرات دنباله را به یک دنباله معتبر تبدیل کنید.

پاسخ:

اگر چند نمونه از گرافهای حاصل از دنباله ها را بررسی کنیم، متوجه میشویم در این گرافها هر مولفه یا دارای ریشه است و یا دارای دور. چرا که به ازای هر راس دقیقا یک یال وجود دارد، اگر یال هیچکس به خودش وارد نشده باشد حتما دور داریم و اگر شده باشد ریشه. نمیتوانیم هر دوی اینها را همزمان داشته باشیم چرا که در این صورت مولفه k راسی حداقل k+1 یال خواهد داشت. همچنین برای یک دنباله معتبر نیاز داریم گراف تبدیل به درخت شود پس باید دور ها را از بین ببریم و مولفه ها را یکی کنیم تا گراف همبند شود. بنابراین میتوانیم اینگونه عمل کنیم، یک مولفه را مولفه نهایی در نظر میگیریم) اگر مولفه ای شامل ریشه وجود داشت همان، در غیر این صورت یکی از مولفه های دور دار به دلخواه(و سپس به ازای هر مولفه شامل دور، یکی از یالهای دور را حذف کرده و به یکی از رئوس مولفه نهایی وصل میکنیم. بنابراین اگر کلا c دور در گراف داشته باشیم، c حرکت برای تبدیل آرایه اولیه به یک آرایه معتبر نیاز داریم. برای پیدا کردن مولفه ها نیز از dfs استفاده میکنیم.

3) درختی با v راس در اختیار داریم. روی هر یک از رئوس عددی طبیعی نوشته شده است. آرایه ای به طول k از زوج مرتب ها مانند (a, b) به عنوان ورودی داده میشود به طوری که a و b اعداد طبیعی هستند. الگوریتمی با پیچیدگی زمانی (O(v+k ارائه دهید که تعیین کند به ازای هر یک از زوج مرتب های (a, b) آیا مسیری از ریشه تا برگ درخت داده شده وجود دارد به طوری که a و b روی مسیر آن باشند یا خیر.

پاسخ: میتوان با اجرای dfs به ازای هر جفت (a, b) چک کرد در مسیر ریشه تا a و تا b، دیگری دیده میشود یا خیر. (o(v*k (نصف نمره) به عنوان راه حل بهینه تر (نمره کامل) میتوان با اجرای یک بار الگوریتم dfs، مقادیر intime (ترتیب دهی به رئوس بر اساس اولین باری که هر یک در پیمایش دیده میشود) و outtime (ترتیب دهی به رئوس بر اساس اخرین باری که هر یک در پیمایش دیده میشود) را به ازای هر راس نگه داشت. سپس به ازای هر جفت (a, b) برای اینکه a و b در مسیری از ریشه تا یک برگ باشند، لازم است همواره یا a در یکی از زیردرخت های b باشد یا بلعکس. برای صحت این موضوع لازم است یکی از دو حالت زیر برقرار باشد:

intime[a] > intime[b] && outtime[a] < outtime[b]

intime[b] > intime[a] && outtime[b] < outtime[a]

بنابراین با یک بار اجرای الگوریتم dfs و محاسبه مقادیر intime و outtime، از o(v) و سپس چک کردن دو شرط موجود برای هر جفت o(k)، مسئله در o(k+v) حل میشود.

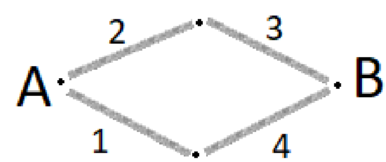
4) صحیح و غلط بودن گزاره های زیر را مشخص کنید. برای گزاره های غلط مثال نقض و صحت بقیه گزاره ها را ثابت کنید.

الف) بررسی وجود دور در گراف در حالت بهینه از O(v) است.

پاسخ: صحیح. اگر گراف دور نداشته باشد از تعدادی مولفه درخت تشکیل شده بنابراین حداکثر v-1 یال دارد. پس با وجود حداقل v یال میتوان وجود دور را نتیجه گرفت.

ب) در یک گراف وزن دار اگر وزن تمام یالها یکتا و مستقل از دیگری باشد، کوتاه ترین مسیبر بین دو راس نیز یکتا است.

پاسخ: غلط. مثال نقض:



ج) اگر یک گراف جهت دار دور نداشته باشد، در پیمایش DFS آن هرگز یال back edge دیده نمیشود.

پاسخ: صحیح. دیده شدن یال back edge معادل وجود دور است چرا که برای مسیری که تا لحظه دیده شدن آن از ریشه طی شده مسیری جهتدار وجود دارد که یال back edge آن را به یکی از رئوس ریشه تا پدرش وصل میکند. لذا دور تکمیل میشود.