۹۸۴۱۱۴۳۲ عرفان زارع

سؤال 1)

walk دنباله شامل رأس‌ها و یال‌هاست که به‌طور متناوب بین رئوس و یال‌های متصل کننده آنها قرار می‌گیرد.

$(v_1 \to v_2) = e_1$ $(v_2 \to v_3) = e_2$ $(v_3 \to v_4) = e_3$ $(v_4 \to v_5) = e_4$ $(v_5 \to v_{10}) = e_5$ $(v_{10} \to v_9) = e_6$ $(v_9 \to v_8) = e_7$

$(v_8 \to v_7) = e_8$ $(v_7 \to v_6) = e_9$ $(v_6 \to v_{10}) = e_{10}$ $(v_1 \to v_6) = e_{11}$ $(v_2 \to v_7) = e_{12}$ $(v_3 \to v_8) = e_{13}$ $(v_4 \to v_9) = e_{14}$

$(v_5 \to v_1) = e_{15}$

walk $= v_1\, e_1\, v_2\, e_2\, v_3\, e_3\, v_4\, e_4\, v_5\, e_5\, v_{10}\, e_6\, v_9\, e_7\, v_8\, e_8\, v_7\, e_9\, v_6\, e_{10}\, v_{10}$

به مجموعه‌ای از یال‌هایی که در جهت حلقه (دور) هستند مجموعه forward arc گوییم. مانند $e_1, e_2, \dots, e_5$ و به مجموعه یال‌های $e_6, \dots, e_{10}$ که در خلاف جهت دور اصلی اند backward arc گوییم. (در خلاف جهت اصلی گراف باشد) به شکل دیگر می‌توان گفت یالی بین v_n و v_m که $n > m$ از v_n به v_m را backward arc گوییم. (برای forward arc برعکس هست)

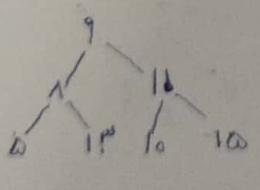
سؤال 2)

در گراف K_n بین هر دو رأس یالی وجود دارد. طبق دانسته‌ها، رئوس در $G(V)$ و $G(V')$ هم یکسان هست که تعداد رئوس آن با K_n یکسان اند. بین مجموعه $G(E)$ و $G(E')$ اشتراکی نیست پس $G(V) + G'(V) \subseteq G(E_{K_n})$. دو رأس v_i و v_j را فرض نمایید. اگر یال میان این دو رأس در G باشد، یعنی در G' یالی میانشان نیست و در $G+G'$ یک یال میانشان است که متعلق به G است و اگر یال میان دو رأس یاد شده در G' باشد، یعنی در G یالی میانشان نیست و در $G+G'$ یک یال میانشان هست که متعلق به G' است.
در واقع به صورت کلی اگر یالی در G باشد، در G' نیست و بالعکس و اگر یالی در G وجود ندارد، باید در گراف کامل وجود داشته باشد.
طبق گفته‌ها و نکات، حال اگر گراف $G+G'$ را در نظر بگیریم، در آن هر رأس مجاور رئوس دیگر، به هم یال داشته و تشکیل گراف کامل را می‌دهند.

در نتیجه $|G(E)| + |G(E')| = |G(E_{K_n})|$

سؤال 3)

نیاز هست با BFS، از ریشه شروع نموده و به ازای هر رأس، فرزند سمت چپ آن، از آن کوچکتر و فرزند سمت راست آن، بزرگتر باشد.
البته باید تمام رئوس زیر درخت چپ یک رأس، از آن کوچکتر و تمام رئوس زیر درخت راست، از آن بزرگتر باشد. برای بررسی این مورد باید در الگوریتممان (BFS) هنگامی که رأسی ویزیت شده با توجه به backtrack بودن الگوریتم، هنگام برگشت به رأس والد، کوچکترین رأس زیر درخت و بزرگترین رأس زیردرخت چپ هر رأس را داشته باشیم که باید هر رأس، از بزرگترین رأس زیردرخت چپ، بزرگتر و از کوچکترین رأس زیردرخت راست، بزرگتر باشد. برای ایجاد مفهوم بهتر، با مثالی بررسی می‌نماییم:



کوچکترین رأس زیردرخت سمت راست ریشه، ۱۰ هست و از ریشه بزرگتره. البته بزرگترین رأس زیردرخت سمت چپ ریشه، ۱۳ است و از ریشه بزرگتره که اینجا دتناقض و شرط ما را برقرار نمی‌کند.

پیچیدگی الگوریتم با توجه به بررسی با BFS از اردر $O(n+m)$ است.

سؤال ۴) دو روش وجود دارد که در ادامه هر کدام را توضیح می دهم. روش 1: با استفاده از BFS روی درخت پیمایش کرده و در ابتدا هر دو مقدار max و min را برابر با مقدار Root قرار می دهیم. سپس در هر پیمایش، اگر مقدار رأس کوچکتر از min بود min را آپدیت و اگر بزرگتر از max بود، مقدار max را آپدیت می کنیم. پیچیدگی حل مسئله هم همچون سؤال قبل و پیچیدگی BFS، برابر $O(n+m)$ است.

روش دوم: با DFS شروع به پیمایش می کنیم و مقدار min را $+\infty$ و مقدار max را $-\infty$ قرار می دهیم. از گره ریشه شروع و هر گره فرزند را به صورت بازگشتی بررسی می کنیم و طبق قواعد اگر وزن گره بیشتر از max بود، max آپدیت و اگر کمتر از min بود، min آپدیت می شود. و به صورت بازگشتی مراحل را طی می کنیم تا تمام گره ها visit شوند. در نهایت خواسته سؤال را در متغیرهای min و max داریم.

سؤال ۵)

می دانیم تعداد تطابق کامل متمایز گراف $k_{n,n}$ برابر با $n!$ است. زیرا برای هر رأس در مجموعه M در مرحله iام، $(n-i)$ رأس برای انتخاب یال از مجموعه N داریم. پس فرمول آن می شود

$$\prod_{i=1}^{n} i = n!$$

برای k_{2n} ولی تعداد تطابق کامل متمایز گراف برابر با مورد زیر می شود. دلیل آن هم آن است که در هر مرحله، ۲ رأس را برای یال matching انتخاب می نماییم. به دلیل تکراری وجود داشتن در حالات، برای جایگشت های معیوب ایجاد تکرار را حذف می کنیم.

$$\frac{\binom{2n}{2}\times\binom{2n-2}{2}\times\binom{2n-4}{2}\times\dots\times\binom{6}{2}\times\binom{4}{2}\times\binom{2}{2}}{n!}=\frac{(2n)!}{2^n\times n!}$$

سؤال ۶)

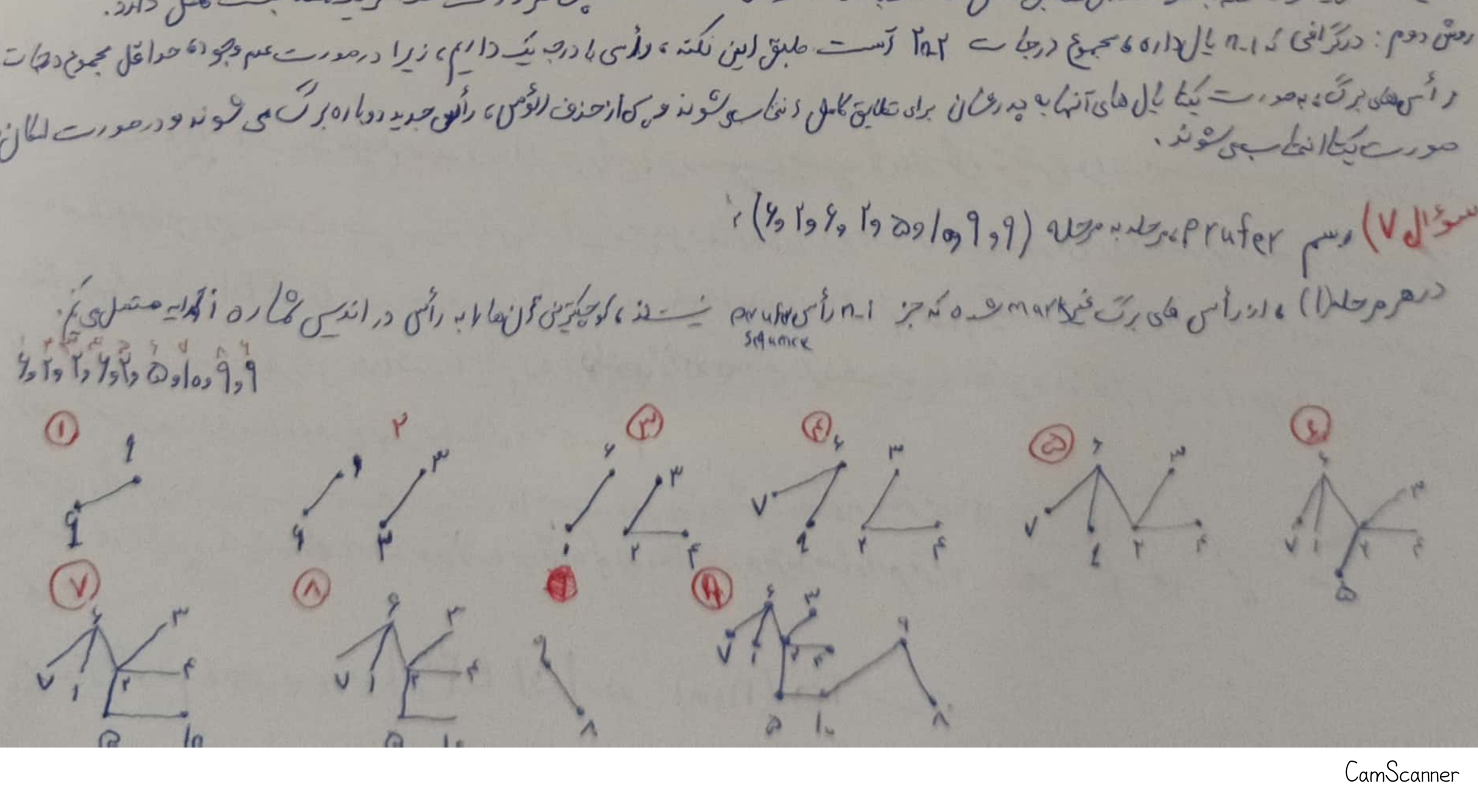
تطابق کامل در یک درخت یعنی همه گره های یک درخت، با گره های مربوطه در درخت دیگر، شکل ساختار و ساختار گره، تطابق دارد. حال برای اثبات فرض می کنیم یک درخت T_1 و دو تطبیق کامل مجزا با درختان T_2 و T_3 داریم. از آنجا که منطبقات کامل متمایز هستند باید حداقل یک گره v وجود داشته باشد که میان T_2 و T_3 متفاوت است. بر روی کمترین جد مشترک (LCA) گره متفاوت در T_2 و T_3 تمرکز کنیم. در LCA، زیر درختی که در آن گره در T_2 ریشه دارد، باید ساختار یا ساختاری متمایز نسبت به زیر درخت مربوطه در T_3 وجود داشته باشد. اگر ساختار یا ساختار متمایز باشند یعنی یکی از درخت ها، اعم از T_2 یا T_3 نمی تواند مطابقت با T_1 داشته باید حداقل یک گره داشته باشد که در آن تطابق کامل متفاوت باشد که با فرض ما در تناقض است. پس هر درخت حداکثر یک مطابقت کامل دارد.

روش دوم: درختی که n-1 یال دارد، مجموع درجات $2n-2$ است. طبق این نکته، رأسی با درجه یک داریم، زیرا در صورت عدم وجود، حداقل مجموع درجات 2n می بود. رأس های برگ، به صورت یکتا یال های آنها به پدرشان برای تطابق کامل انتخاب می شوند و پس از حذف آن رأس، رأس جدید دوباره برگ می شوند و در صورت امکان به صورت یکتا انتخاب می شوند.

سؤال ۷) رسم prufer، مرحله به مرحله (۹، ۹، ۱۰، ۵، ۱۰، ۲، ۶، ۲، ۶، ۶):

در هر مرحله (i)، از رأس های برگ، فعلی max عدد n-1 رأس prufer sequence نیستند، کوچکترین آن ها با رأس در اندیس شماره i آرایه متصل می کنیم.

۹، ۹، ۱۰، ۵، ۱۰، ۶، ۲، ۶، ۲، ۶



سوال ۸) ماکسیمم تعداد یال یک گراف دو بخشی با دو مجموعه A دارای ۷ رأس و B دارای ۱۰ رأس باشد، تعداد یال گراف $K_{7,10}$ است. که می‌شود ۷۰ = ۱۰ × ۷. بله رابطه دارند و حداکثر تعداد یال های گراف دو بخشی با A و B رأس برابر $||K_{A,B}||$ که همان $|A| \times |B|$ است.