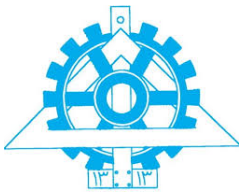

به نام خداوند بخشنده مهربان

تحلیل و طراحی الگوریتم ها، نیم سال اول سال تحصیلی ۹۸-۹۷

پاسخ تمرین چهارم

۲۹ آذر ۱۳۹۷

---

۱. از آنجایی که رابطه زیر برقرار است:

$$d(u,v) = d(u,w)\ xor\ d(w,v)$$

و ریشه بین هر دو مسیر دلخواه بین دو راس دلخواه u و v وجود دارد، پس با یک بار اجرای الگوریتم dfs میتوان فواصل ریشه از تمام رئوس را به دست آورد$(O(m))$ و سپس با استفاده از رابطه بالا، فاصله n جفت راس را به دست آورد$(O(n))$.

۲. گراف G را به گونه ای میسازیم که از هر راس G، دو کپی گرفته شده است که یکی از آنها برای زمان های زوج و یکی برای زمان های فرد است. هر یال (u,v) در گراف G در صورتی که زمان زوج باشد، از u به v و در صورتی که زمان فرد باشد، از v به u خواهد بود. بنابراین، تبدیل این یال در گراف G به صورت یک یال بدون جهت (u_{even}, v_{odd}) خواهد بود.
در صورتی که در گراف G در یک راس (مثلا u) بمانیم، در G در واقع از u_{even} به u_{odd} یا برعکس آن حرکت کردیم. در نتیجه، باقی ماندن در یک خانه به صورت یک یال بدون جهت بین u_{even} و u_{odd} خواهد بود.
حال پس از ساختن G'، الگوریتم bfs را از راس 1_{even} (با توجه به این که از راس 1 در زمان 0 حرکتمان را شروع میکنیم) اجرا میکنیم. با توجه به بدون وزن بودن یال ها، این الگوریتم کوتاهترین مسیر بین راس 1_{even} و سایر رئوس را محاسبه خواهد کرد. کوتاهترین مسیر به راس n ام برابر با کوتاهترین مسیر به راس n_{even} یا n_{odd} میباشد.

۳. گرافی دوبخشی است که بتوان آن را با 2 رنگ، رنگ آمیزی کرد. بنابراین، با استفاده از الگوریتم bfs، گراف را رنگ آمیزی میکنیم(در هر سطح، رنگ فرزندان یک راس را مخالف رنگ پدرش قرار میدهیم). در صورتی که یالی بین 2 راس همرنگ وجود نداشته باشد، گراف دوبخشی است.
این الگوریتم برای تشخیص گراف n بخشی قابل استفاده نیست. زیرا مشخص نیست که در هر سطح از الگوریتم bfs، فرزندان آن عضو کدام بخش هستند و در واقع، رنگشان چیست.

۴. با ساده سازی صورت این مسئله، در می یابیم که در واقع به دنبال وجود دور منفی در گراف شهرها از راس 1 هستیم. تنها نیاز به ایجاد تغییر در وزن ها متناسب با وجود $chest$ در آنها(در صورت وجود $epic\ chest$ بین 2 راس i و j، $W(i,j) = s(i,j)-2$ و در صورت وجود $rare\ chest$، $W(i,j) = s(i,j)-1$ و در غیر اینصورت $W(i,j) = s(i,j)$) داریم.
برای یافتن دور منفی، کافیست $n-1$ بار، تمام یال ها را $relax$ کنیم. در نهایت، یک بار دیگر عمل $relaxation$ را انجام میدهیم. در صورتی که فاصله بین 2 راس تغییر کند، بنابراین دور منفی خواهیم داشت. پس کافیست راس شروع را یکی از راس های موجود در این دور در نظر بگیریم و این دور را دائما پیمایش کنیم.

۵. گراف G را به صورت زیر ایجاد میکنیم: هر حرف، یک راس در این گراف است و به ازای هر دو رشته i و j، اگر حرف آخر i با حرف اول j یکسان بود، یک یال از i به j ایجاد میکنیم.
حال باید بررسی کنیم که آیا دوری در این گراف وجود دارد که از تمام یال ها بگذرد (دور اویلری). برای این کار، 2 شرط زیر را باید بررسی کنیم:

آ) درجه ورودی و خروجی هر راس با یکدیگر برابر باشند.

ب) گراف مورد نظر، قویا همبند باشد. یعنی از هر مولفه ی همبندی، به هر مولفه همبندی دیگری مسیری وجود داشته باشد و بالعکس (راس هایی که یالی ندارند در نظر گرفته نمیشوند)
شرط اول با پیمایش تمام راس ها و بررسی تعداد یال های ورودی و خروجی، بررسی خواهد شد. شرط دوم با یکبار اجرای dfs روی گراف و بررسی این که آیا تمام رئوس در یک مولفه همبندی هستند یا خیر، بررسی میشود.

بدین ترتیب، در صورتی که هر دو شرط بالا برقرار باشد، دوری با شرایط گفته شده وجود خواهد داشت.

۶. برای حل این سوال، از الگوریتم BFS استفاده می‌کنیم. هر راس در گراف مورد نظر بیانگر یک عدد طلایی است و اگر عدد این راس x باشد، اعداد رئوس همسایه‌ی آن برابر با $x0$ و $x1$ هستند.
در ابتدا عدد 1 را به عنوان راس شروع در نظر می‌گیریم و در صف اضافه می‌کنیم. در هر قدم، یکی از صف برمیداریم(مثلا y) و $y0$ و $y1$ را به انتهای صف اضافه می‌کنیم. در هر قدم، بررسی می‌کنیم که آیا عدد مورد بررسی بر n بخش پذیر هست یا نه. در صورتی که بخش پذیر باشد، این عدد همان پاسخ مورد نظر است. چون در هر سطح از این الگوریتم، اعداد به ترتیب بزرگ و بزرگتر می‌شوند، بنابراین اولین عددی که بر n بخش پذیر باشد، کوچکترین عدد ممکن نیز هست.