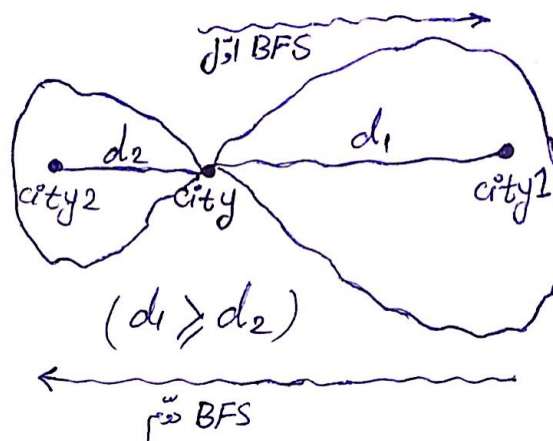




مسئله‌ی ۱. قتل بی‌رحمانه بنا

رای هر شهر یک $node$ در نظر بگیرید و در آن شماره شهرهایی که می‌توان از آن شهر به آن‌ها رفت ($adj[]$) و همچنین یک مقدار $boolean$ برای این که شهر دارای قسمتی از جسد هست یا نه ($hasPart$) را نگه می‌داریم.

حال از یک شهر دلخواه BFS می‌زنیم و دورترین شهری را از این شهر که $hasPart$ آن $true$ است، بدست می‌آوریم. ($city1$) سپس از شهر بدست آمده دوباره BFS می‌زنیم و دورترین شهری که از این شهر بیشترین فاصله را داراست و دارای تکه‌ای از جسد است بدست می‌آوریم. ($city2$)



شکل ۱: نحوه یافتن دورترین شهرهای دارای قسمتی از جسد

دقت کنید که با BFS اول دورترین شهری دارای تکه‌ای از جسد که در یک سمت $city$ می‌باشد یافت می‌شود. (با توجه به شکل راست‌ترین شهر در واقع بدست می‌آید)

$city1$ و $city2$ دورترین شهرهایی از هم هستند که تکه‌ای از جسد را دارند. حال هر شهری که از $city1$ و $city2$ فاصله‌ای کمتر از d داشته باشد، می‌تواند بنا در آن شهر منفجر شده باشد و تعداد این شهرها خروجی می‌دهیم. (دقت کنید با BFS دوم فاصله‌ی سایر رئوس تا $city1$ بدست آمده است. برای بدست آوردن فاصله رئوس تا $city2$ باید BFS دیگری از این راس بزنیم)

مسئله‌ی ۲. رویای محمود

هنگام زدن dfs روی یک گراف جهت‌دار، اگر یالی از راس فعلی به راسی که هنوز در پشته بازگشتی ازش خارج نشده‌ایم وجود داشته باشد، دور جهت‌دار داریم. می‌توان با نگه داشتن پدر هر راس،

یال‌های سازنده این دور را یافت. پس با زدن dfs بر روی تمامی رئوس یک گراف جهت‌دار تا وقتی که همه آنها دیده شوند، در صورت وجود می‌توان دورهای جهت‌دار را یافت.

برای حل این سوال می‌توان هر یال را یکبار برداشت و بررسی کرد که آیا گراف باقی‌مانده دور جهت‌داری دارد یا نه. اما این راه حل از $O((n+m)m)$ است و پذیرفته نمی‌شود. پس باید یال‌هایی که حذف می‌شوند را کاهش داد.

همان طور که در بالا گفته شد، با زدن dfs بر روی تمامی رئوس دیده نشده، دورهای جهت‌دار در صورت وجود پیدا می‌شوند. پس اگر یک دور را در نظر بگیریم، می‌توان فقط یال‌هایی که در این دور وجود دارند را یکبار حذف کرد و بررسی کرد که آیا گراف باقی‌مانده دور جهت‌داری دارد یا خیر. این راه حل از $O((n+m)n)$ است و پذیرفته می‌شود.

مسئله‌ی ۳. داستان سمی

مسئله بیان شده را به یک مسئله گراف تبدیل می‌کنیم به این صورت که هر کانال را به صورت یک گره از گراف در نظر می‌گیریم همچنین گره “-” را به گراف اضافه می‌کنیم و با شروط زیر یال‌هایی را بین این گره‌ها رسم می‌کنیم:

- اگر دکمه up سالم بود، از گره کانال i به گره $i + 1$ یال با وزن ۱ وصل می‌کنیم.
- اگر دکمه $down$ سالم بود از گره i به $i - 1$ یالی با وزن ۱ رسم می‌کنیم.
- اگر دکمه “-” سالم بود از تمامی گره‌ها به این راس یالی با وزن ۱ وصل می‌کنیم و این راس را به گره‌هایی با عدد دو رقمی با یالی با وزن ۲ متصل می‌کنیم. (علت وزن ۲ آن است که باید ۲ دکمه بعد از “-” برای رفتن به کانال‌های دو رقمی زده شود).
- به هر کدام از گره‌ها با کانال یک رقمی اگر دکمه‌اش سالم بود از سایر کانال یال با وزن ۱ به این گره‌ها وصل می‌کنیم.

حال کافی است از گره کانال X ، الگوریتم $dijkstra$ را اجرا کنیم و مقدار کوتاه‌ترین فاصله‌ای که از گره X تا راس Y بدست می‌آید را خروجی دهیم.