

## ساختمان داده‌ها و الگوریتم‌ها

### کویز چهارم - گراف

شماره دانشجویی:

نام و نام خانوادگی:

تاریخ: ۱۴۰۳/۳/۲۰

۳۰ نمره

۱.

اثبات کنید از هر نود دلخواهی مانند  $u$  در یک درخت الگوریتم DFS را اجرا کنیم. در نهایت به یک سر قطر درخت میرسیم (منظور از قطر درخت، طولانی ترین path ممکن در درخت است)

پاسخ:

با استفاده از برهان خلف فرض میکنیم که نود مقصد یکی از سرهای قطر نیست،

پس اقدام به نام گذاری میکنیم:

$u$ : نود دلخواه مبدا

$v$ : نود مقصد

$d, d'$ : دو سر قطر واقعی

نود  $w$ : اولین نودی در قطر که از  $u$  میرسیم (نزدیک ترین نود در قطر به  $u$ )

اگر که نود  $v$  هیچ یک از نودهای  $d, d'$  نباشد:

دو حالت پیش میاید:

۱. از نود  $u$  ما به نود میرسیم  $w$  اما از آنجا به یک نود  $v'$  میرسیم که  $v' \neq v$  این بدین معنی است که ما به یک مسیر طولانی تر (قطر بزرگتر) راه یافته ایم. که نشان دهنده ی تناقض است چون میدانیم که دوسر قطر  $d, d'$  هستند.

۲. فرض کنید به  $w$  نرسیدیم. بنابراین:

$$\text{len}(u, v) \geq \text{len}(u, w) + \max(\text{len}(w, d), \text{len}(w, d'))$$

$$\text{len}(w, v) \geq \max(\text{len}(w, d), \text{len}(w, d'))$$

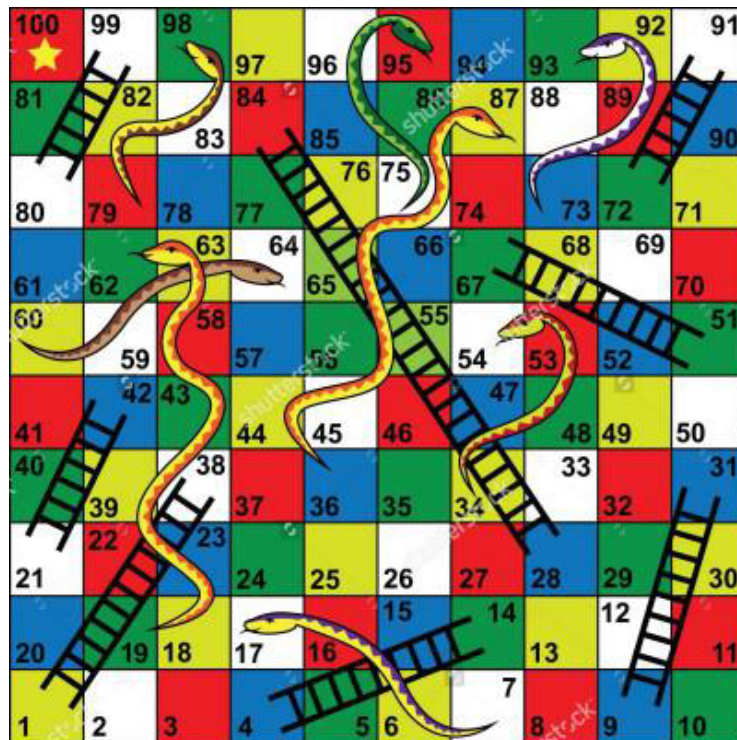
که در این صورت هم به تناقض میرسیم.

۲.

۳۰ نمره

الگوریتمی ارائه دهید که به کمک گراف، بتوان کمترین مقدار لازم پرتاب تاس برای برنده شدن در یک صفحه بازی مار و پله را تشخیص داد.

به عنوان مثال در مورد زیر حداقل ۷ پرتاب تاس لازم است تا بتوان بازی را تمام کرد:



پاسخ:

ایده این است که تخته مار و پله را به عنوان یک گراف جهت‌دار در نظر بگیریم و جستجوی اول سطح (BFS) را از نقطه شروع، گره ۱، مطابق با قوانین بازی انجام دهیم. ما یک گراف جهت‌دار ایجاد می‌کنیم، با در نظر گرفتن شرایط زیر:

۱. برای هر رأس در گراف،  $v$  ما یک یال از  $v$  به  $v+1, v+2, v+3, v+4, v+5, v+6$  داریم زیرا می‌توانیم با یک پرتاب تاس از گره  $v$  به هر یک از این گره‌ها برسیم.

۲. اگر هر یک از این همسایگان  $v$  یک نردبان یا مار داشته باشد که ما را به موقعیت  $x$  می‌برد، در این صورت  $x$  به جای پایه نردبان یا سر مار، همسایه می‌شود.

اکنون مسئله به یافتن کوتاه‌ترین مسیر بین دو گره در یک مسئله گراف جهت‌دار کاهش می‌یابد.

۳.

۴۰ نمره

یک درخت با  $V$  گره و  $E$  یال در اختیار داریم که هر کدام از گره‌های آن یک عدد طبیعی است. آرایه‌ای به طول  $k$  از زوج مرتب‌ها مانند  $(a, b)$  به عنوان ورودی داده می‌شوند که  $a$  و  $b$  اعداد طبیعی هستند. الگوریتمی با پیچیدگی زمانی  $O(V + E + k)$  ارائه دهید که تعیین کند به ازای هر کدام از زوج مرتب‌های  $(a, b)$ ، آیا مسیری از ریشه تا برگ درخت وجود دارد به طوری که  $a$  و  $b$  روی آن مسیر باشند یا خیر.

پاسخ:

درخت مد نظر را یک بار با الگوریتم  $DFS$  پیمایش می‌کنیم و به ازای هر گره زمان ورود به گره ( $discoveryTime$ ) و زمان خروج از گره ( $departureTime$ ) را ذخیره می‌کنیم. حال برای هر زوج مرتب  $(a, b)$  اگر یکی از دو شرط زیر برقرار باشد،  $a$  و  $b$  روی یک مسیر قرار دارند:

$$discoveryTime[a] < discoveryTime[b] \text{ and } departureTime[a] > departureTime[b]$$

$$discoveryTime[b] < discoveryTime[a] \text{ and } departureTime[b] > departureTime[a]$$

اگر شرط اول برقرار باشد،  $a$  به ریشه نزدیک‌تر است و برای شرط دوم برعکس.

بنابراین با هزینه  $O(V + E)$  با الگوریتم  $DFS$  درخت را پیمایش کردیم و زمان‌های ورود و خروج را ذخیره کردیم. حال با هزینه  $O(k)$ ، به ازای هر زوج مرتب، شروط بالا را بررسی می‌کنیم. بنابراین هزینه کل برابر است با:  $O(V + E + k)$