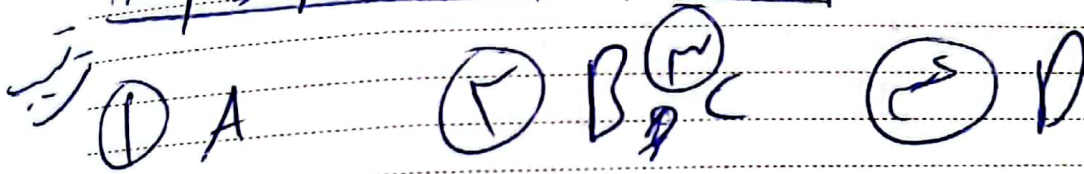


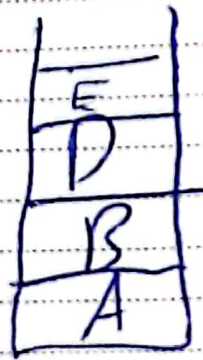
①

Bfs:

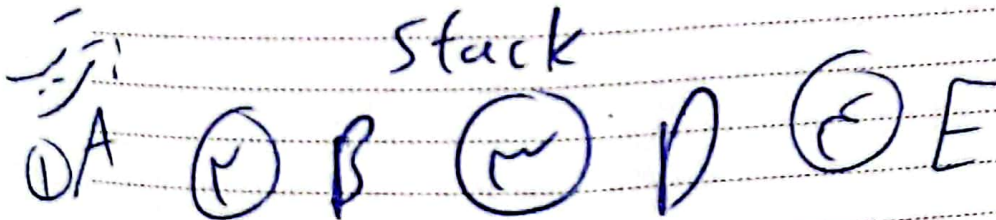


مسیر: A B D G    A C D G

Dfs:



Stack



مسیر: A B D E G

(۲) در این روش ما  $Dfs$  و  $Bfs$  را با هم ترکیب  
می‌کنیم و ابتدا  $Dfs$  را با حقوق یک می‌زنیم و این حقوق را  
برای  $S$  می‌گذاریم و اگر جواب در آن نبود هر بار حقوق را یکی  
بیشتر می‌کنیم.

مزیت این روش این است که نسبت به  $Bfs$  حافظه  
کمتری دارد (مصرف می‌کند) ولی همزمان به خاطر بررسی چند بار  
بعضی از حقوق‌ها تا جواب از  $Bfs$  کمتر است.  
اما این زمان اضافه خیلی زیاد نیست و از  $S-1$  تا  $S$   
است. و این نسبت به کل معادلات ماضی زیاد نیست.

در واقع سرعت خوبی در حد  $Bfs$  دارد و  
نسبت به  $Bfs$  چون در بار اول شروع می‌کنیم  
و چیزی را نگه‌داری نمی‌کنیم و حافظه‌ای نداریم  
 $Dfs$  دارد.

March				
Mon.	7	14	21	28
Tue.	1	8	15	22
Wed.	2	9	16	23
Thu.	3	10	17	24
Fri.	4	11	18	25
Sat.	5	12	19	26
Sun.	6	13	20	27



در این مسئله حالت اولیه Node است که از آن شروع  
می‌کنیم و حالت نهایی Node است که به آن می‌رسیم  
رسم و برای هر Node هم حالت با حرکت دادن  
هر خانه در خانه خالی به وجود می‌آید (بالا، پایین  
چپ، راست)

بنابر این می‌توانیم حالات این Puzzle را از شروع  
تا پایان به شکل یک درخت در آورده و روی  
آن DFS یا BFS زده  
در DFS چون به دنبال چپ‌ترین جواب می‌رویم  
و درخت ما تا بی‌نهایت ادامه دارد و ممکن است  
هرگز به جواب نرسیم  
در BFS نزدیک‌ترین جواب به Root  
پیدا می‌شود

فروردین												
۲۷	۲۰	۱۳	۶	شنبه								
۲۸	۲۱	۱۴	۷	شنبه								
۲۹	۲۲	۱۵	۸	۱	شنبه							
۳۰	۲۳	۱۶	۹	۲	شنبه							
۳۱	۲۴	۱۷	۱۰	۳	شنبه							
	۲۵	۱۸	۱۱	۴	شنبه							
	۲۶	۱۹	۱۲	۵	جمعه							

یکشنبه  
Sunday

۱۴

فروردین ۱۴۰۱  
3 April 2022  
۱ رمضان ۱۴۴۳

ROOYAN  
DARZI  
PHARMACEUTICAL COMPANY

که این مسئله با در نظر گرفتن هزینه برای  
تست Node و در نظر گرفتن این موضوع  
که Node که قبلاً فرزندانش بر روی سرهانه  
نیاز دوباره به تستش ندارد بهتر است  
حل می شود.

در این سوال نیز راهنمای سایت  
geek for geek استفاده شده است.

April			
Mon.	4	11	18 25
Tue.	5	12	19 26
Wed.	6	13	20 27
Thu.	7	14	21 28
Fri.	1	8	15 22 29
Sat.	2	9	16 23 30
Sun.	3	10	17 24