

# کاشی‌های چرخان

تکلیف پیاده سازی اول

۱۸ آبان



مدرس:  
دکتر شمس فرد  
دستیاران آموزشی:  
آرش اعتماد  
صالح جعفریزاده  
یاسمن روحانی‌فر

# پایه

Name = A Gr = 1	Name = B Gr = 1	Name = C Gr = 2	Name = D Gr = 2
Name = L Gr = 1	Empty Gr = 1,5	Empty Gr = 2,5	Name = E Gr = 2
Name = K Gr = 3	Empty Gr = 3,5	Empty Gr = 4,5	Name = F Gr = 4
Name = J Gr = 3	Name = I Gr = 3	Name = H Gr = 4	Name = G Gr = 4

Figure 1

شکل ۱

❖ جدولی ۴ در ۴ با ۱۲ کاشی و ۴ خانه خالی مطابق شکل ۱ در نظر بگیرید. هر خانه متعلق به یک یا دو گروه است.

❖ به عنوان مثال در شکل داده شده  $Gr = 1,5$  به معنای آن است که آن خانه متعلق به گروه‌های ۱ و ۵ است.

❖ در هر مرحله مجاز به انجام یکی از اقدام‌های زیر هستید.

- (۱) چرخش ۹۰ درجه گروه اول در جهت عقربه‌های ساعت
- (۲) چرخش ۹۰ درجه گروه دوم در جهت عقربه‌های ساعت
- (۳) چرخش ۹۰ درجه گروه سوم در جهت عقربه‌های ساعت
- (۴) چرخش ۹۰ درجه گروه چهارم در جهت عقربه‌های ساعت
- (۵) چرخش ۹۰ درجه گروه پنجم در جهت عقربه‌های ساعت

Name = A Gr = 1	Name = B Gr = 1	Name = C Gr = 2	Name = D Gr = 2
Name = L Gr = 1	Empty Gr = 1,5	Empty Gr = 2,5	Name = E Gr = 2
Name = K Gr = 3	Empty Gr = 3,5	Empty Gr = 4,5	Name = F Gr = 4
Name = J Gr = 3	Name = I Gr = 3	Name = H Gr = 4	Name = G Gr = 4

Figure 1



Name = L Gr = 1	Name = A Gr = 1	Name = C Gr = 2	Name = D Gr = 2
Empty Gr = 1	Name = B Gr = 1,5	Empty Gr = 2,5	Name = E Gr = 2
Name = K Gr = 3	Empty Gr = 3,5	Empty Gr = 4,5	Name = F Gr = 4
Name = J Gr = 3	Name = I Gr = 3	Name = H Gr = 4	Name = G Gr = 4

Figure 2



Name = L Gr = 1	Name = A Gr = 1	Name = C Gr = 2	Name = D Gr = 2
Empty Gr = 1	Empty Gr = 1,5	Name = B Gr = 2,5	Name = E Gr = 2
Name = K Gr = 3	Empty Gr = 3,5	Empty Gr = 4,5	Name = F Gr = 4
Name = J Gr = 3	Name = I Gr = 3	Name = H Gr = 4	Name = G Gr = 4

Figure 3

# مسئله

- ❖ در ابتدا یک حالت تصادفی از چیدمان کاشی ها در جدول به شما داده می‌شود. ترتیب مناسبی از اقدام‌های مجاز پیدا کنید تا با اعمال آن به روی جدول داده‌شده، جدول اولیه (شکل ۱) به دست آید.
- ❖ پیاده‌سازی را با استفاده از الگوریتم BFS انجام دهید.
- ❖ برنامه خود را با مثال های داده شده تست کنید. (مثال ها بعدا آپلود می‌شود) و خروجی را طبق فرمت خواسته شده در فایل متنی ذخیره نمایید.
- ❖ پیاده‌سازی را با استفاده از الگوریتم  $A^*$  انجام دهید.
- ❖ تعریف تابع Heuristic به عهده خودتان است. (Heuristic بهتر امتیاز بیشتر!)
- ❖ برنامه خود را با مثال های داده شده تست کنید. (مثال ها بعدا آپلود می‌شود) و خروجی را طبق فرمت خواسته شده در فایل متنی ذخیره نمایید.
- ❖ جواب های این مرحله را با مرحله قبل مقایسه کنید.

حداکثر فضای حافظه ای که در اختیار برنامه خود برای اجرا می‌گذارید  $2G$  و حد اکثر زمانی که نیاز است برای جواب منتظر بمانید ۵ دقیقه می‌باشد. (در صورتی که برنامه برای رسیدن به منابعی بیشتر از این نیاز داشت در سطر مربوط در فایل متنی عبارت No Answer را چاپ نمایید.)

# ساختار ورودی و خروجی برنامه

## ❖ ساختار ورودی

❖ یک رشته به طول ۱۶ از کاراکتر هایی که بیان کننده نام هر کاشی هستند. ('N' برای کاشی‌های خالی استفاده می شود.)

## ❖ ساختار خروجی

❖ در سطر اول خروجی تعداد اقدام لازم برای رسیدن به حالت هدف و در هر سطر بعد به ترتیب شماره گروه هایی که چرخش باید روی آنها انجام شود تا جواب برسیم.

❖ مثال (برای زمانی که ورودی مانند شکل داده شده باشد)

### ❖ ورودی:

LACDNBNEKNNFJIHG

### ❖ خروجی:

3

1

1

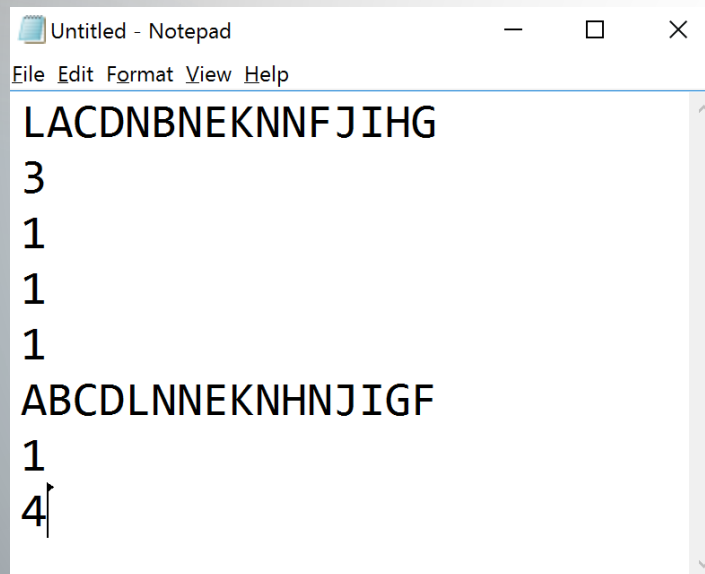
1

Name = L Gr = 1	Name = A Gr = 1	Name = C Gr = 2	Name = D Gr = 2
Empty Gr = 1	Name = B Gr = 1,5	Empty Gr = 2,5	Name = E Gr = 2,5
Name = K Gr = 3	Empty Gr = 3,5	Empty Gr = 4,5	Name = F Gr = 4
Name = J Gr = 3	Name = I Gr = 3	Name = H Gr = 4	Name = G Gr = 4

Figure 2

# ساختار فایل متنی خروجی

- ❖ به ازای هر بار اجرای برنامه در سطر اول رشته ای که برنامه را برای آن اجرا کرده اید و در سطر های بعد عینا جواب خروجی را درج میکنید.
- ❖ یک فایل برای اجرای BFS و یک فایل برای اجرای  $A^*$
- ❖ مثال:



```

Untitled - Notepad
File Edit Format View Help
LACDNBNEKNNFJIHG
3
1
1
1
ABCDLNNEKNHNJIGF
1
4
    
```



# تحویل دادنی های پروژه

- ❖ Source Code و فایل متنی خروجی خواسته شده
- ❖ الزامی بر admissible بودن تابع heuristic نیست.

## ❖ موارد امتیازی:

- ❖ تعریف و استفاده از admissible heuristic (به همراه اثبات رسمی یا غیر رسمی آن)
- ❖ هر سند مفیدی که نتیجه گرفته شده از این پروژه در آن شرح داده شده باشد.

موفق باشید!