

تکلیف کامپیوتری شماره ی ۱

خلاصه

در این تمرین می‌خواهیم یک عامل عقلایی با استفاده از الگوریتم‌های جست و جو برای بازی ۸-پازل یا Sliding Puzzle بنویسیم. به این منظور یک فایل jar به عنوان سرور در اختیار شما قرار داده شده‌است. شما باید کد کلاینت را به گونه‌ای کامل کنید تا در هر نوبت با توجه به نتیجه‌ی جست و جوی شما یکی از ۴ دستور {Up, Down, Left, Right} را به سرور ارسال کند، سرور با اعمال تغییر روی صفحه‌ی بازی نتیجه را برای شما ارسال خواهد کرد. بازی تا رسیدن صفحه‌ی بازی به حالت مطلوب ادامه دارد.

سرور

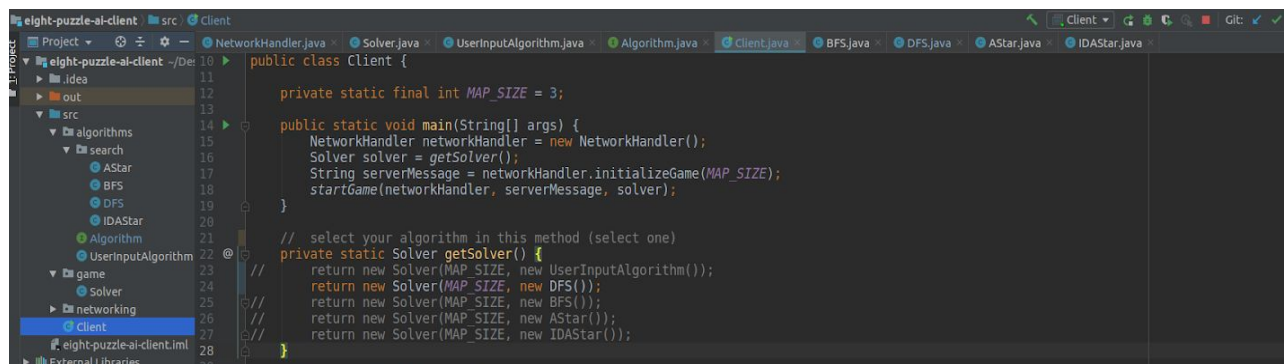
یک فایل jar که با ران کردن آن از طریق دستور `java -jar eight-puzzle-ai-server.jar` در ترمینال اجرا می‌شود و دستورهای لازم را به شما نشان می‌دهد. همچنین می‌توانید با دادن آرگمان `input` در ترمینال صفحه‌ی بازی را دستی وارد کنید (به طور پیش‌فرض صفحه رندوم پر میشود):

> `java -jar eight-puzzle-ai-server.jar input`

```
amirali@amirali-GL502VSK:~$ cd Desktop/
amirali@amirali-GL502VSK:~/Desktop$ java -jar eight-puzzle-ai-server.jar input
Server is running, waiting for client to connect...
Client Connected...
enter board initial state:
2      3      4
4      6      7
8      1      0
board initialized.
█
```

کلاینت

یک پروژه‌ی ناقص است که به کمک شما کامل می‌شود. برای این کار لازم است تابع `makeMove` را در کلاس‌های DFS, BFS, AStar, IDAStar کامل کنید. این توابع در هر نوبت بازی صفحه را در قالب یک آرایه‌ی ۲بعدی ورودی می‌گیرند و باید به عنوان خروجی یکی از ۴ رشته‌ی {"Up", "Down", "Left", "Right"} را برگردانند. شما در کلاس `Client` و در تابع `getSolver` می‌توانید الگوریتم سرچ خود را مانند نمونه انتخاب کنید.



توجّهات

۱. ابتدا سرور و سپس کلاینت را ران کنید.

۲. برای DFS حداکثر تا عمق ۳۰۰ را مشاهده کنید

۳. باقی قسمت‌های کلاینت مربوط به اتصال به سرور است

۴. سورس کدهای سرور و کلاینت را می‌توانید در لینک‌های زیر ببینید:

- Client: <https://github.com/monjar/eight-puzzle-ai-client>
- Server: <https://github.com/monjar/eight-puzzle-ai-server>

۵. خواندن کدهای سرور ممکن است به شما در نوشتن بعضی توابع کمک کند.