تمرین دوم – جستجوی مسیر بهینه بر روی گراف

مسئله: در یک فضای 3×3 مقادیر ۱، ۲، .... و ۸ به همراه یک فضای خالی قرار دارند. با هر حرکت می‌توان جای فضای خالی و یکی از (حداکثر) چهار خانه کنار آن را با هم تعویض کرد. هدف در این بازی رسیدن از یک چینش مشخص به چینش مشخص دیگری است. در این مسئله قصد داریم با استفاده از الگوریتم A^* این معما را حل کنیم.

الف) در این مسئله حالت‌ها، عمل‌ها و آزمون هدف را به طور دقیق تعریف کنید.

ب) تابعی بنویسید که با گرفتن یک حالت مشخص، حالت‌های بعدی (فرزندها را) را تولید کند. آیا از یک حالت مشخص می‌توان به هر حالت دلخواه مسیری داشت؟ توضیح دهید.

ج) تابعی بنویسید که فاصله Manhattan بین دو حالت را محاسبه نماید.

د) با استفاده از توابعی که در قسمت‌های ب و ج نوشته‌اید الگوریتم جستجوی A^* را (با مقدار Heuristic فاصله Manhattan) برای 8-Puzzle پیاده‌سازی نمایید.

ه) با اجرای الگوریتم نوشته شده نشان دهید که برای رفتن از حالت $\begin{bmatrix} 7 & 2 & 8 \\ 4 & 5 & 6 \\ 3 & & 1 \end{bmatrix}$ به حالت $\begin{bmatrix} 1 & 2 & 3 \\ 7 & 6 & 5 \\ 8 & & 4 \end{bmatrix}$ حداقل ۲۶ حرکت لازم است.

و) یک Heuristic مناسب‌تر می‌تواند جمع فاصله Manhattan و دو برابر تعداد تضادهای خطی موجود (Linear Conflicts) باشد. در مقایسه دو حالت مشخص، دو خانه x و y دارای تضاد خطی هستند اگر در هر دو حالت این دو خانه در یک سطر یا ستون یکسان بوده ولی در یکی از حالت‌ها x قبل از y در آن سطر یا ستون مشخص ظاهر شود و در دیگری y قبل از x در همان سطر یا ستون ظاهر شود. الگوریتم جستجوی A^* را با این Heuristic برای 8-Puzzle پیاده‌سازی نمایید و آن را برای حالت‌های شروع و پایان قسمت **ه** اجرا نمایید.

ز) پیچیدگی فضایی و زمانی دو قسمت **ه** و **و** را با هم مقایسه نمایید. آیا استفاده از تعداد گره‌های اضافه‌شونده به Q معیار مناسبی برای مقایسه پیچیدگی زمانی است دو قسمت **ه** و **و** است؟ توضیح دهید.

توجه: یک راه ساده (ولی غیر بهینه) نمایش هر حالت با یک عدد ۳۲ بیتی است. این کار با استفاده از نمایش محل هر یک از خانه‌های ۱ الی ۸ در جدول با ۴ بیت امکان‌پذیر است. دستورات de2bi و bi2de در MATLAB می‌توانند در این راه مفید واقع شوند.