



- مهلت ارسال پاسخ تا ساعت ۲۳:۵۹ روز مشخص شده است.
- همکاری و همفکری شما در انجام تمرین مانعی ندارد اما پاسخ ارسالی هر کس حتما باید توسط خود او نوشته شده باشد.
- در صورت همفکری و یا استفاده از هر منابع خارج درسی، نام همفکران و آدرس منابع مورد استفاده برای حل سوال مورد نظر را ذکر کنید.
- لطفا تصویری واضح از پاسخ سوالات نظری بارگذاری کنید. در غیر این صورت پاسخ شما تصحیح نخواهد شد.

سوالات عملی (۱۰۰+۱۰ نمره)

۱. (۳۰ نمره) به جویتر رجوع کنید.
 ۲. (۳۵+۱۰ نمره) احتمالا تاکنون نام بازی چند وزیر (N Queens) را شنیده‌اید. بازی به این شکل است که باید n وزیر شطرنج را در یک صفحه $n * n$ طوری قرار دهید که به صورت قطری و سطری و ستونی هم‌دیگر را تهدید نکنند. در این سوال باید مسئله‌ی چند وزیر را با استفاده از الگوریتم‌های Constraint Satisfaction Problem که در طول درس با آن آشنا شده‌اید حل کنید. با استفاده از $Backtracking$ چینش مناسبی برای وزیرها پیدا کنید به گونه‌ای که هیچ دو وزیری در یک سطر، یک ستون یا اریب قرار نگیرند. برای اینکه بتوانید در زمان محدود پاسخ مسئله برای n های بزرگتر را بیابید می‌توانید از هیوریستیک‌های معرفی شده در درس مانند ($ForwardChecking$ و ...) استفاده کنید.
- نکات قابل توجه:
- در فایل تکستی که پیوست تمرین شده‌است، اندازه‌ی ورودی‌های سوال قرار داده شده‌است.
 - کد شما به ازای هر ورودی حدودا یک دقیقه فرصت محاسبه‌ی پاسخ را دارد.
 - زمان اجرای کد خود به ازای هر ورودی را در ستون مقابل آن بنویسید.
 - اگر زمان اجرا بیشتر از یک دقیقه شد، * قرار دهید.
 - برای رعایت عدالت در نمره‌دهی تنها مجاز به استفاده از زبان پایتون هستید.
 - سعی کنید با تحلیل ویژگی‌های مسئله، هیوریستیک‌های مناسب برای حل سوال را بیابید.
 - در صورتی که کدتان بتواند برای ورودی‌های ۷۰۰ و ۸۰۰ خروجی مناسب را در زمان معقول ایجاد کند تا ده نمره امتیازی دریافت می‌کنید.

ورودی: عدد n

خروجی: یک ماتریس $n * n$ به طوری که برای خانه‌های فاقد وزیر - و برای خانه‌های دارای وزیر ۰ قرار داده شده است.

نمونه‌ی ورودی:

۴

نمونه‌ی خروجی:

- 0 - -
 - - - 0
 0 - - -
 - - 0 -

۳. (۳۵ نمره) در این سوال می‌خواهیم یک بازی را با روش $min - max$ حل کنیم. این بازی بین ۲ نفر است. یک جدول مربعی داریم و ۱ مهره از هر دو نفر در نقاطی در جدول قرار داده شده‌است. در ادامه بازیکنان باید به نوبت مهره‌های خود را به جدول اضافه کنند. بازی تا چند مرحله پیش می‌رود و در انتها فردی که امتیاز بیشتری بدست بیاورد برنده است.

قوانین قرار دادن مهره: در هر مرحله فقط در مجاورت مهره‌های قبلی خودمان می‌توانیم مهره قرار دهیم. (در نمونه کد قرار داده شده برای کاهش فضای جست‌وجو این مورد محدودتر شده است.)

معیار امتیازدهی: هر کدام از مهره‌هایی که در زمین داریم امتیازی به ما می‌دهند. اینکه چه اندازه امتیاز به ما بدهند به این بستگی دارد که در اطراف آن مهره، در کدام موقعیت‌ها مهره‌های دیگری از ما وجود داشته باشد. مثلاً در نمونه کد قرار داده شده، اگر مهره دیگری در موقعیت ۲- بالا- ۲- راست قرار داشته باشد ۱۰۰۰ نمره به ما تعلق می‌گیرد و

در فایل‌های قرارداده شده کلیت بازی و توابع امتیازدهی و ... پیاده شده‌اند. شما صرفاً باید $min - max$ را در دو حالت $original$ و $pruning$ پیاده کنید. کد حالت رندوم نیز در اختیار شما قرار گرفته‌است.

شما باید ۱۱ حالت مختلف از بازی را اجرا کرده و موارد زیر را در گزارش تحویل دهید.

خروجی (شامل زمان‌های اجرا) هر ۱۱ حالت، نمودارهای هر ۱۱ حالت، تحلیل از دلیل باخت یا برد و تحلیل زمانی (مقایسه بین حالت‌های $original$ و $pruning$).

برای این سوال باید ۳ فایل `minmax_original.py` و `minmax_pruning.py` و گزارش را آپلود کنید.