

به نام خدا



درس هوش مصنوعی و سیستم‌های خبره

---

## تمرین سری چهارم

---

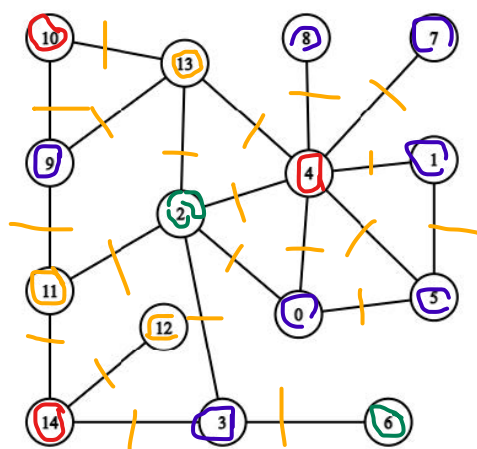
طراحان: محمد یارمقدم، امیرعلی پاکدامن  
مدرس درس: جناب آقای دکتر محمدی

مهلت ارسال: ۱۴۰۱/۰۸/۰۷

مثلا شاید با سه تا بشه ولی ما بگیم چهار

بخش تئوری

۱. با توجه به اینکه رنگ آمیزی گراف يك مسئله NP-complete مي باشد، بررسی درست بودن يك پاسخ براي مسئله آسان مي باشد ولي پيدا كردن پاسخ مورد نظر كاري دشوار است. در اين تمرين سعي مي كنيم براي اين مسئله يك پاسخ بهينه به كمك الگوريتم ژنتيك پيدا كنيم. (يا حداقل يك upper bound خوبي براي مسئله) گراف زير را در نظر بگيريد. مي خواهيم با كمترين تعداد رنگ راس هاي گراف را به گونه اي رنگ كنيم كه هيچ دو راس مجاور هم رنگ نباشند.



optimization

local search

genetic

Encoding technique

(gene, chromosome)

توليد جمعيت اوليه

در ابتدا با توجه به مسئله و گراف مد نظر جمعيت اوليه را تشكيل دهيد. نحوه توليد جمعيت و encoding خود را به طور كامل توضيح دهيد. (فرض كنيد اندازه جمعيت اوليه برابر با ۶ است).

Initialization procedure (creation)

محاسبه Fitness

نحوه محاسبه fitness هر کروموزم (يك عضو جمعيت) را بيان كنيد. با محاسبه fitness value مربوط به هر عضو از جمعيت، اعضا را به ترتيب fit بودن، مرتب كنيد. اين مقادير چه چيزي را نشان مي دهند؟

Evaluation function

Crossover و Mutation

در اين مرحله عمليات توليد نسل جديد را انجام دهيد و با ارائه روش هاي crossover و mutation روي جمعيت اوليه اجرا كنيد و نسل جديد را بدست آوريد.

Genetic operators (mutation, recombination)

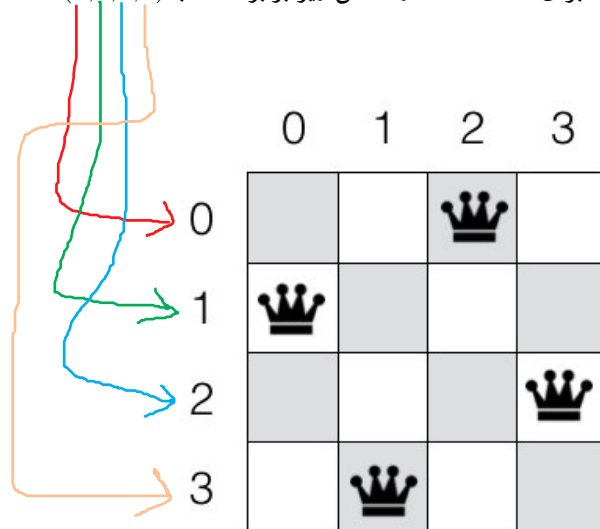
Selection of parents (reproduction)

نسل جديد

با توجه به آنچه كه در قسمت قبل انجام داديد، اكنون بايد يك جمعيت جديد داشته باشيد. مقادير Fitness را براي اعضاي اين جمعيت محاسبه كنيد. سپس مجموع Fitness جمعيت فعلي و رابا مجموع Fitness جمعيت مرحله قبل مقايسه كنيد. نتيجه اين مقايسه چه چيزي را نشان مي دهد؟

## بخش عملی:

۱. پیاده سازی روش حل مسئله  $n$  وزیر به روش hill-climbing در مسئله  $n$  وزیر می خواهیم  $n$  وزیر را در یک صفحه شطرنج  $n$  در  $n$  بچینیم به گونه ای که هیچ دو وزیر یکدیگر را تهدید نکنند. شما باید دو فایل `NQueens.py` و `hill_climbing.py` را تکمیل کنید. در فایل اول باید مدل مسئله را پیاده کنید. با کامنت وظیفه هر تابع و گاری که باید انجام دهد مشخص شده است. برای سهولت در پیاده سازی state های این مسئله را tuple ای به طول  $n$  در نظر بگیرید که هر عضو آن عددی از صفر تا  $n-1$  است. به طوری که `state[i]` شماره ستون وزیر مستقر در سطر  $i$  را بیان میکند. در واقع فرض میشود در هر سطر تنها یک وزیر حضور دارد. برای مثال state در عکس زیر برابر است با (2,0,3,1)



در فایل دوم نیز باید تابع `hill_climbing` را با توجه به الگوریتم آن پیاده سازی کنید. این تابع به عنوان ورودی یک شی از جنس `NQueens` خواهد گرفت و از توابع آن استفاده میکند. در نهایت میتوانید با اجرای فایل `main.py` الگوریتم خود را بسنجید. توجه داشته باشید که این الگوریتم همیشه در همه حالات جواب درست بدست نخواهد داد. برای اطمینان از کارایی کدتان، کد در ازای  $n$  های ۴ و ۵ خروجی ۱۰/۱۰ خواهد داد. ولی ملاک نمره دهی کد پیاده سازی شده است.

## قوانین:

۱. تمرین ها به صورت فردی انجام شوند و حل گروهی تمرین ها مجاز نیست.
۲. برای تحویل تمرین یک فایل zip شامل فایل اولیه تغییر داده شده توسط خودتان، با نام [HW4\_ID\_NAME] در سامانه gradescope بارگذاری کنید.