

به نام خدا



دانشگاه صنعتی امیرکبیر

(پلی تکنیک تهران)

دانشکده مهندسی کامپیوتر و فناوری اطلاعات

گزارش پروژه اول درس مبانی و کاربردهای هوش مصنوعی

عنوان پروژه

پیاده سازی الگوریتم های جستجو

نگارش

نرگس سدیفی - ۹۷۳۱۰۹۰

استاد درس

جناب دکتر بهنام روشن فکر

آذر ۱۳۹۹

چکیده

هدف پروژه ، بررسی ، تحلیل و پیاده سازی ۳ الگوریتم جستجو بر روی مسئله ی دسته بندی کارت ها است.

واژه های کلیدی:

الگوریتم A^* ، الگوریتم BFS ، الگوریتم IDS

مقدمه

در این پروژه ، تمرکز اصلی بر پیاده سازی، تحلیل و بررسی ۳ الگوریتم زیر در قالب ۳ سوال در مساله ای است که در صورت پروژه مطرح شده:

۱. الگوریتم BFS

۲. الگوریتم IDS

۳. الگوریتم A*

نحوه ی اجرا

این سه سوال در سه فایل با نام های $q1$, $q2$, $q3$ قرار دارند و برای اجرا به کلاس اضافه ای نیاز ندارند. برای اجرای هر فایل کافی تابع `main` درون آن را اجرا کنید

فرمت ورودی

فرمت ورودی برای تمامی سوال ها مشابه صورت پروژه است

فرمت خروجی

در سطر اول جواب نهایی سپس عبارت **GOAL!** چاپ می شود و پس از آن عمق جواب تعداد نود های تولید شده و بسط داده شده چاپ می شوند

آزمون هدف

اگر تمامی کارت های یک رنگ در یک دسته و به صورت نزولی مرتب شده باشند به هدف رسیده ایم.

شرح انجام پروژه

سوال اول

در سوال اول به پیاده سازی الگوریتم BFS پرداخته شده است.

در این برنامه یک صف frontier و explored داریم نود شروع را در لیست frontier قرار می دهیم سپس تا وقتی به هدف نرسیدیم نود ها را از لیست فرانتیر برداشته و در اکسپلورد قرار می دهیم و در هر مرحله پس از بررسی نود های موجود در یک عمق سراغ فرزندان آنها می رویم

سوال دوم

در سوال دوم به پیاده سازی الگوریتم IDS پرداخته شده است. در این الگوریتم ابتدا در یک تابع روش جستجوی عمق اول با عمق محدود پیاده سازی می شود. برنامه با ورودی گرفتن عمق ابتدایی با شروع از آن عمق و تا بی نهایت هر بار برای هر عمق، تابع جستجوی عمق اول با عمق محدود را صدا میکند. در صورتی که پاسخ پیدا شود برنامه متوقف، و مسیر از گره اولیه تا گره پاسخ به همراه عمق جواب چاپ میشود و در غیر اینصورت افزایش عمق و اجرای مجدد تا ابد ادامه خواهد داشت.

سوال سوم

در این سوال علاوه بر تابع اصلی الگوریتم، به طراحی دو تابع دیگر برای به دست آوردن H , G می پردازیم.

تابع $function_g()$:

در این تابع به بررسی هزینه ی حقیقی ای که تا الان پرداخت شده تا به نود کنونی برسیم، پرداختیم

تابع $function_h()$: در این تابع به بررسی هزینه تخمینی ای که تا نود هدف مانده می پردازیم این

هیوریستیک به این صورت در نظر گرفته شده که در ابتدا چک میکند آیا عدد اول هر ردیف برابر با n (تعداد اعداد) هست یا خیر اگر نبود به اعضای تمام کارت های بعدی یک عدد به هیوریستیک مان اضافه می کنیم در غیر این صورت به بررسی رنگ کارت های بعدی میپردازیم به اعزای هر کارت با رنگ اشتباه نیز یک

عدد به هیوریستیک اضافه میکنیم و در صورت یکسان بودن رنگ ها نزولی بودن عدد ها را چک میکنیم و
هرجا این شرط نقض شده بود نیز یک عدد به هیوریستیک مان اضافه میکنیم از آن جا که هزینه ی تابع
تخمینی همواره کمتر از هزینه ی واقعی نود است پس قابل قبول است

مقایسه ی الگوریتم ها

در تست کیس های امتحان شده عمق جواب برای همه ی الگوریتم ها یکسان به دست آمد
تعداد نود های تولید شده و گسترش داده شده در الگوریتم BFS از همه بیشتر پس از در IDS و در A^*
کمترین تعداد نود گسترش شده و تولید شده را داشتیم.