

## گزارش پروژه اول درس هوش مصنوعی: محمدرضا صادقیان 9731121

### بخش اول) پیدا کردن یک نقطه ثابت غذا با استفاده از جستجوی اول عمق(3 امتیاز)

**سوال:** آیا ترتیب کاوش همان ترتیبی بود که انتظار داشتید؟ آیا پکمن در راه در راه رسیدن به هدف، به همه مربع های کاوش شده می رود؟

ترتیب کاوش شده دقیقا ترتیب مورد انتظار بود یعنی از یک گره شروع کرده و عمیق می شود تا به جواب برسد و اگر جوابی نبود به لایه بالایی برگشته و مسیر دیگر را می پیماید. خیر به همه مربع های کاوش شده نمی رود و دلیل آن این است که در DFS در ابتدا تمام محاسبات را انجام می دهیم و سپس مسیر پیدا شده را پیمایش می کنیم و در این بخش هم چند مسیر وجود دارد که کاوش شده است ولی چون هدف نبوده اند و در نتیجه در مسیر نهایی نیستند بنابراین آنها پیمایش نمی شوند.

**سوال:** آیا این راهحل کمترین هزینه را دارد؟ اگر نه فکر میکنید که جستجوی اول عمق چه کاری را اشتباه انجام میدهد؟

این راه کمترین هزینه را ندارد و میتوان راههای بهینه تر نیز پیشنهاد داد. مشکل جستجوی اول عمق این است که از ابتدا عمق را پیش میرود و هر جا که به خانه هدف میرسیم جستجو را متوقف میکند و این نتیجه می شود که ما شانس بررسی مسیرهای بهتر را از دست بدهیم چون وقتی در عمق حرکت میکنیم امکان دارد در همان شاخه اول به خانه هدف برسیم. همچنین امکان دارد که در پیمودن عمق خانه هایی که نیاز نباشد را هم پیماییم صرفا چون در مسیرمان در پیمایش عمق قرار دارد در صورتی که پیمودن آن منطقی نیست.

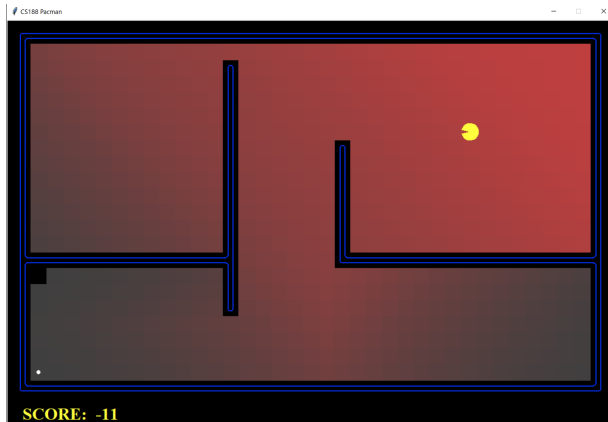
### بخش چهارم) جستجوی A استار(3 امتیاز)

**سوال:** الگوریتمهای جستجویی که تا به این مرحله پیاده سازی کرده اید را روی openMaze اجرا کنید و توضیح دهید چه اتفاقی می افتد.

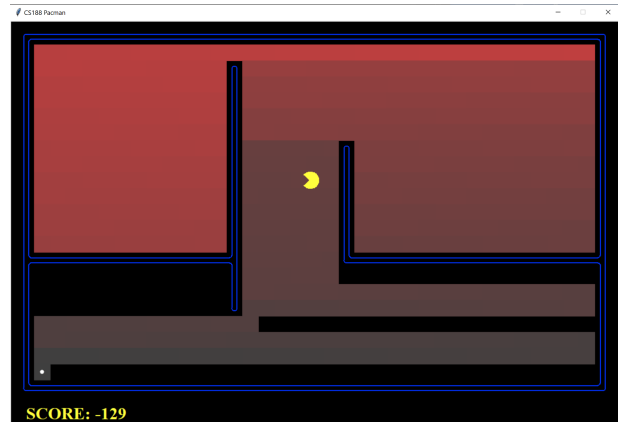
الگوریتم DFS خیلی مسیر بدی را انتخاب میکند زیرا خط به خط چک می کند و همچنین از آنجا که قسمت سمت چپ مسیر رنگ روشن تری داری نشان می دهد که این بخش را اول بررسی کرده است که خوب واضح است که کار بیهوده و غلطی می باشد. البته چون بخش سیاه بیشتری نسبت به BFS و UCS دارد یعنی گره های کمتری گسترش یافته اند زیرا عمق بررسی کرده است و هدف نیستند.

سه الگوریتم بعدی مسیر یکسانی را انتخاب می کنند که بهینه ترین راه است؛ ولی این که  $A^*$  قسمت سیاه بیشتری دارد نشان از این موضوع است که گره های کمتری گسترش یافته اند.

BFS



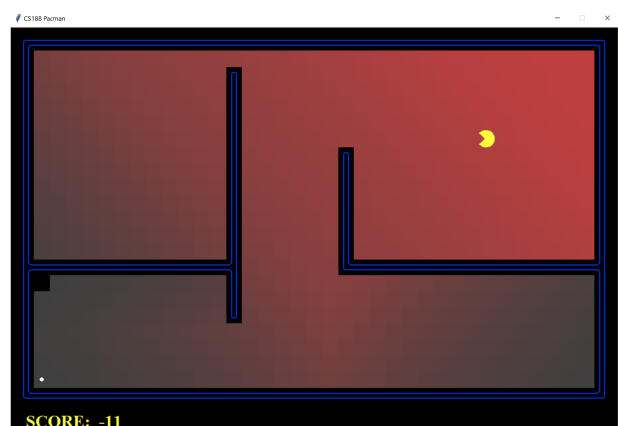
DFS



$A^*$



UCS



### بخش ششم) هیوریستیک برای مسئله گوشه ها(3 امتیاز)

**سوال:** هیوریستیک خود را توضیح دهید و سازگاری آن را استدلال کنید.

یک لیست از اندیس Corner هایی که visit نشده اند (از تمامی corner های موجود در state) تشکیل می دهیم. در یک حلقه هر بار فاصله ی منتهن از موقعیت فعلی تا هر corner ای که visit نشده را حساب می کنیم و در لیست دیگری ذخیره می کنیم. حال بین فواصل منتهن محاسبه شده مینیم می گیریم و اگر از مینیم فاصله ی اولیه کمتر بود به عنوان مینیم فاصله ی اولیه در نظرش می گیریم و اندیس corner ای که فاصله ی منتهنش را به عنوان مینیم اولیه در نظر گرفتیم را از لیست corner های visit نشده پیدا می کنیم و آن عضو لیست (که در حقیقت اندیس آن corner در لیست تمامی corner های موجود در state است) را به عنوان اندیس نزدیکترین corner ذخیره می کنیم. حال موقعیت فعلی را برابر با corner متناظر با نزدیک ترین corner قرار می دهیم و مینیم فاصله ی محاسبه شده را به مسافت کل اضافه می کنیم در نهایت اندیس نزدیک ترین corner را از آرایه ی اندیس corner های visit نشده حذف می کنیم. هنگامی که لیست اندیس corner های visit نشده خالی شد از حلقه خارج می شویم و مسافت کل را برمی گردانیم. هیوریستیک در اینجا سازگار و قابل قبول است زیرا از فاصله ی منتهن استفاده می کند و دیوار ها را در نظر نمی گیرد در نتیجه قطعا از هزینه ی واقعی کمتر است یا با آن برابر است همچنین در هر بار اجرای حلقه مینیم فاصله ی محاسبه شده به مسافت کل اضافه می شود و corner های visit شده حذف می شوند (مقدار  $F_n$  در طول مسیر صعودی است و با انجام هر عمل مقدار هیوریستیک حداکثر به اندازه ی هزینه ی آن عمل کاهش می یابد).

### بخش هفتم) خوردن همه نقطه ها(4 امتیاز)

**سوال:** هیوریستیک خود را توضیح دهید و سازگاری آن را استدلال کنید.

هیوریستیک ما به این صورت عمل می کند که در ابتدا حالت اولیه را می گیرد و Boolean محل غذاها را نیز می گیرد. بعد از آن به متغیر هزینه مان مقدار اولیه صفر را می دهیم. حال بر روی آرایه دو بعدی Boolean غذاهایمان پیمایش می کنیم و در هر نقطه چک می کنیم که آیا غذا هست و آیا maze Distance مربوط به آن نقطه از هزینه ای که تا این لحظه داشتیم بیشتر است یا نه و اگر بیشتر بود آن را به جای مقدار قبلی متغیر هزینه قرار می دهیم. maze Distance به این صورت عمل می کند که الگوریتم BFS را برای رفتن از مبدا تا خانه هدفمان پیاده سازی می کند که در واقع در اینجا مختصات خانه هدف، مختصات خانه غذایمان است. بعد از اجرای الگوریتم جستجوی اول سطح طول مسیر پیموده شده را به ما برمیگرداند. حال فلسفه هیوریستیک ما این است که یک به یک خانه های غذا را بررسی می کند و در نهایت طول مسیر پیموده شده برای دورترین مسیر توسط الگوریتم BFS را به عنوان هیوریستیک ما برمیگرداند.

حال سازگار بودن هیوریستیک مان را باید بررسی کنیم. ابتدا شرط اول سازگاری را بررسی می کنیم و اثبات می کنیم که هیوریستیک حالت های هدفمان برابر با صفر است. وقتی که به حالت هدف می رسیم یعنی اینکه همه خانه های غذا خورده شده اند در نتیجه وقتی می خواهیم هیوریستیک را محاسبه کنیم چون همه خانه های غذا False هستند پس در نتیجه وارد شرط if داخل حلقه مان نمی شوند در نتیجه مقدار هزینه همان مقدار اولیه اش که برابر صفر است را به عنوان مقدار هیوریستیک حالت هدف برمیگرداند که این مقدار صفر شرط اول سازگاری را برقرار می کند. حال به سراغ شرط دوم میرویم. فرض می کنیم که يك گره والد و يك گره فرزند داریم و از آنجایی که گره والد و فرزند خانه های مجاور یکدیگر هستند هزینه واقعی پیمایش بینشان برابر با يك واحد می شود. حال فرض می کنیم که هیوریستیک را برای گره فرزند به دست آورده ایم. می دانیم که maze Distance براساس الگوریتم جستجوی اول سطح عمل می کند که هرگاه هزینه حرکت بین گره های همسایه همگی برابر يك باشد بهینه است که در اینجا نیز به همین صورت است در نتیجه هیوریستیک ما سعی دارد هزینه بهینه رسیدن به دورترین نقطه غذا را به عنوان هیوریستیک به ما بدهد. حال هیوریستیک مربوط به گره فرزند را داریم. حال ما ادعا می کنیم که دورترین خانه غذا برای گره والد، همان خانه غذایی است که برای گره فرزند دورترین است (که در این حالت تفاوت هیوریستیک برابر با يك می شود) و یا خانه غذای دیگری است که هیوریستیک به دست آمده از آن با هیوریستیک به دست آمده از گره فرزند برابر است. دلیلی که ما برای این استدلالمان داریم این است که ما از گره فرزند به والد هم مسیر داریم. اگر دورترین خانه غذا برای گره والدمان (A) با دورترین خانه غذا برای گره فرزندمان (B) متفاوت بود و هیوریستیک مربوط گره والدمان بزرگتر از گره فرزندمان بود در آن صورت چون از گره فرزند به گره والد نیز مسیر داشتیم در آن صورت دورترین خانه برای گره فرزندمان هم باید همان A می شد. همچنین با توجه به اینکه BFS سطح به سطح بررسی می کند بنابراین گره والد در مسیر جستجوی اش مسیر گره فرزند را نیز می پیماید. نتیجه گیری کلی با توجه به مطالب بالا این است که تفاوت هیوریستیک ها برای گره والد و فرزند یا برابر صفر است یا برابر يك که در هر دو حالت با هزینه واقعی پیمایش گره والد به فرزند، کوچکتر مساوی است در نتیجه شرط دوم سازگاری نیز برقرار شد.

### بخش هشتم) جستجوی نیمه بهینه (3 امتیاز)

**سوال:** برای مثال ممکن است در مواقعی رفتن به نزدیکترین نقطه منجر به این شود که یک مسیر را چندین بار طی کنیم درحالیکه پس از پیمایش اول دیگر غذایی در آن مسیر باقی نمانده در نتیجه راه حل غیر بهینه می شود. (مثلا به جای اینکه یک مسیری را انتخاب کند که بدون اینکه لازم باشد آن را چند بار طی کند تمامی غذا ها را بخورد به طور حریصانه سمت نزدیکترین غذا برود که باعث شود بعضی مسیر ها را چند بار طی کند تا بتواند همه ی نقاط را بخورد.)