گزارش پروژه اول هوش مصنوعی

حسین بهشتی فرد (9631011)

در تمامی سوال ها یک آرایه path برای نگه داشتن مسیر ها تعریف شده است که به هنگام تولید گره های فرزند متناسب با جایگاه فرزند(چپ، بالا، راست، پایین) پر می شود.

سوال 1)

IDS: نحوه ی کار این الگوریتم به این صورت است که در هر مرحله یک limit برای عمق در نظر می گیرد و تا آن عمق الگوریتم DFS را اجرا می کند. در صورت سوال گفته به صورت افزایش تدریجی باشد یعنی از عمق ۱ شروع کرده و در هر مرحله الگوریتم DFSتا آن عمق اجرا می شود و سپس عمق یکی زیاد می شود و دوباره الگوریتم از اول اجرا می شود تا زمانی که به هدف برسیم.

برای پیاده سازی مسئله یک حلقه بی نهایت درنظر گرفته شده که از عمق 1 شروع کرده و در هر مرحله الگوریتم DFS را متناسب با عمق فراخوانی می کند و سپس عمق را یکی زیاد می کند تا زمانی که به هدف برسد.

آزمون هدف نیز به این صورت پیاده سازی شده که اگر در هر صف شماره کلاس متفاوت بود و یا ترتیب قد نزولی نبود و یا "#" در سر صف نبود،False و در غیر این صورتTrue بر می گرداند.

که هنگام بسط هر گره آزمون انجام می شود و در صورت رسیدن به هدف حلقه به اتمام می رسد.

نمونه ورودی و خروجی:

INPUT:

100a 120a 110a

120b # 110b

OUTPUT:

depth = 10

up

left

down

right

up

right

down

left

up

left

number of nodes is = 15338

number of expanded nodes is = 15328

سوال ۲)

جست و جوی A*: این الگوریتم بر خلاف دو سوال دیگر آگاهانه است و در هر مرحله سعی می کند که بهترین انتخاب را انجام دهد. که این انتخاب به این صورت است که در هر مرحله حالتی از لیست frontier **انتخاب می شود که کمترین هزینه (هزینه ای که تابع هیوریستیک می دهد + عمق) را دارد.**

برای تابع هیوریستیک پیشنهادی، مینیموم مجموع فاصله (عمودی + افقی) هر فرد درصف، با جایگاهش در حالت نهایی در بین تمامی حالات نهایی است. یعنی اگر به عنوان مثال فرد k (یکی از n فرد داخل صف ها) در خانه (x_k,y_k) باشد، و در حالت نهایی ام در خانه (xi_k,yi_k) قرار گرفته است، برای هر حالت نهایی، مقدار زیر را محاسبه میکنیم

$$\sum_{k=1}^{n} |yik - yk| + |xik - xk|$$

حال در تمام حالات نهایی این مقدار را چک میکنیم و کمترین را به عنوان هیوریستیک بر میگردانیم. تا بهترین راه تا نزدیک ترین هدف پیدا کنیم.

هیوریستیک ما قابل قبول است زیرا به طول کلی نتیجه ای کمتر از هزینه اصلی ارائه میدهد و به نوعی خوشبینانه است. همچنین چون تمام حالات نهایی را برای هر گره چک میکند، میتواند بهترین راه را پیدا کند و بهترین نتیجه را ارائه دهد.

به همین خاطر الگوریتم ما تنها وابسته به یک حالت نهایی نیست و همیشه سعی بر پیدا کردن بهترین حالت نهایی و جواب را دارد.

برای پیاده سازی یک آرایه costs **در نظر گرفته شده که به هنگام تولید گره ها به ازای هر گره تابع هیوریستیک فراخوانی می شود و هزینه ای که می دهد با عمق آن گره جمع شده و درون** index **متناظر با گره در آرایه** costs **ریخته می شود.**

آزمون هدف نیز در هنگام بسط هر گره چک می شود و به اینصورت است بررسی می کند آیا آن گره برابر با یکی از حالات هدف است یا خیر.

نمونه ورودی و خروجی 1:

INPUT:

2 3

100a 120a 110a

120b # 110b

OUTPUT:

depth = 10

up

left

down

right

up

right

down

left

up

left

number of nodes is = 61

number of expanded nodes is = 47

نمونه ورودی و خروجی 2:

INPUT:

4 3

160a 170b 165c

150b 180d 165b

178d 160a

180a 160c 175d

OUTPUT:

depth = 26

right

up

left

up

right

right

down

left

left

down

right

right

down

left

up

up

up

right

down

down

down

left

up

up

left

down

number of nodes is = 6149

number of expanded nodes is = 3331

سوال ۳)

جست و جوی دو طرفه: این الگوریتم به این صورت است که الگوریتم BFS هم از سمت حالت شروع و هم از سمت هدف اجرا می شود تا به یک حالت مشترک برسند که در واقع یعنی به هدف رسیده ایم.

برای پیاده سازی این مسئله دو تا آرایه frontier و visited برای حالت شروع و دو آرایه دیگر برای حالت هدف در نظر گرفته شده است که در ابتدا آرایه ورودی درون frontier حالت شروع وارد می شود و تمامی حالت های هدف درون frontier اهداف ریخته می شوند و یک حلقه بی نهایت داریم که در هر مرحله یک بار bfs را برای حالت شروع (فقط بسط یک نود) و بار دیگر از طرف هدف فراخوانی می کند تا زمانی که به هم برسند.

آزمون هدف نیز به این صورت است که در هر مرحله و برای هر طرف چک می شود اگر آن گره ای که بسط داده می شود در frontier طرف مقابل باشد هست یا خیر که اگر وجود داشته باشد یعنی به هدف رسیده ایم.

نمونه ورودی و خروجی:

INPUT:

2 3

100a 120a 110a

120b # 110b

OUTPUT:

depth = 10

up

left

down

right

up

right

down

left

up

left

number of nodes is = 85

number of expanded nodes is = 62

مقایسه الگوریتم ها:

از نظر عمق جواب یکسان هستند زیرا در الگوریتم سوال اول چون به تدریج عمق زیاد می شود در نتیجه جوابی را در نظر می گیرد که در کمترین عمق است و در سوال دوم نیز در هر مرحله گره ای را بسط می دهد که ما را به نزدیک ترین هدف می رساند و در سوال سوم تمامی گره های هدف در frontier **قرار گرفته که در نتیجه به نزدیک ترین هدف می رسد.**

از نظر تعداد گره ها همانطور که در خروجی ها میبینیم بیشترین تعداد را الگوریتم ids **دارد زیرا ناآگاهانه است و در هر عمق از اول دوباره گره های تکراری تولید می کند زیرا حافظه ای برای نگهداری ندارد و کمترین تعداد را الگوریتم** A* **دارد زیرا آگاهانه است و گره هایی را بسط می دهد که ما را به هدف نزدیک می کند.**

از نظر گره های بسط داده شده چون ids **به صورت افزایش عمق تدریجی در نظر گرفته شده در نتیجه اکثر(به جز گره های عمق آخر که بعد از حالت هدف قرار گرفته اند) گره های تولید شده بسط داده می شوند و در جست و جوی** A* **چون آگاهانه است و بهترین را انتخاب و بسط می دهد و در جست و جوی دو طرفه چون حالت هدف را حالتی در نظر گرفتیم که به تمامی حالت های هدف متصل است میبینیم که تعداد زیادی از گره ها بسط داده نشده اند.**