۱) State:

وضعیت نقشه را به عنوان State درنظر گرفته ام. به این صورت که پوزیشنی از غذاها و تعداد آن ها و مکان agentها را نگه داشته ام و درصورت تغییر هرکدام از این ها وارد یک State جدید میشوم.

Initial state:

مکان اولیه غذاها و تعداد آن ها و مکان اولیه agentها.

Action:

هرکدام از Agentها میتوانند به چهار طرف بالا و پایین و چپ و راست بروند(به شرطی که درآن طرف غذای زهردار یا دیوار یا آنیکی ایجنت نباشد و استیت قبلا دیده نشده باشد)

Performance measure:

پیدا کردن جواب در کمترین عمق در سریعترین زمان ممکن.

Goal:

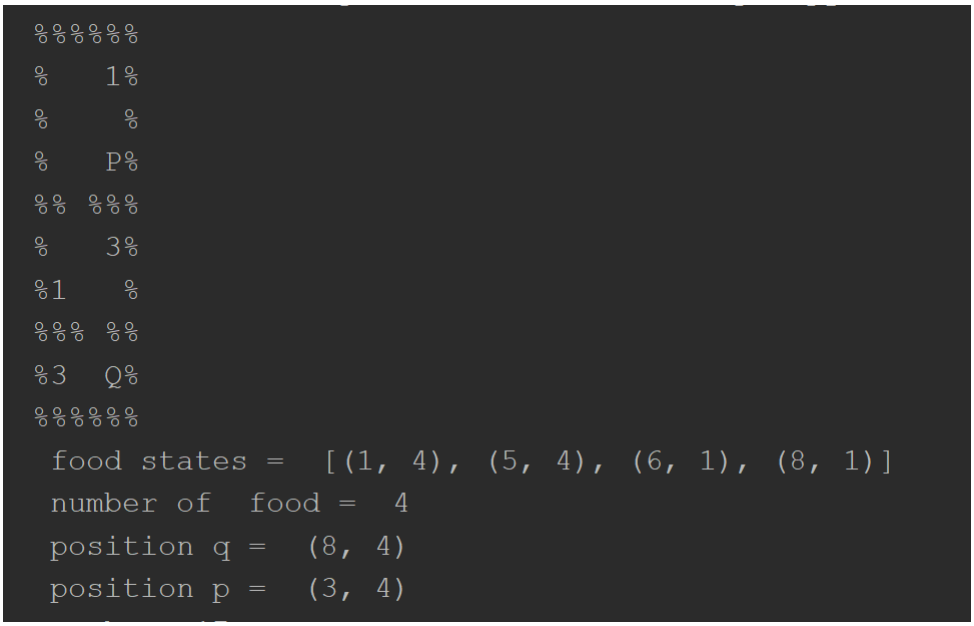
خوردن همه غذاها به طوری که هر Agent غذایی که میتواند را بخورد.

Path cost:

هر جابجایی agentها یکی به هزینه اضافه میکند.

۲) Bfs : ابتدا عامل هوشمند p حرکت خود را به ۴طرف انجام میدهد (اگر امکان پذیر باشد) و به ازای هر حرکت به استیت جدید میرود (اگر این state قبلا دیده شده باشد در صف وارد نمیشود. و اگر دیده نشده باشد به آخر صف اضافه میشود.) سپس q نیز همینگونه حرکت خود را انجام میدهد.
مثلا در نقشه زیر اولین state در پایین نقشه نشان داده شده است.
حرکت دوم: p یا به سمت چپ میرود یا بالا و q به سمت چپ میرود.(۳ تا استیت گسترش میابند) همه این استیت ها به علاوه استیت اولیه در explored_stateها ذخیره میشوند(۴ استیت ذخیره شده اند) و State اولیه از صف منتظر بازدید حذف میشود و این حرکت ها به ترتیب گفته شده به صف اضافه

میشوند. درنتیجه در حرکت بعدی p که درمکان (۳و۳) قرار دارد میتواند به بالا برود یا چپ یا q که درمکان (۴و۸) قراردارد به مکان چپ برود و... ولی نمیتواند به استیت قبلی برود.



Ids: در این الگوریتم از عمق ۱ شروع به پیدا کردن جواب میکنم و اگر پیدا نشد عمق را زیاد میکنم.(یکی یکی) فرق اساسی این الگوریتم با bfs این است که Stateهای مرزی که دیده میشوند در اول صف انتظار وارد میشوند و نه در آخر آن. و هر بار تا یک عمق برای پیدا کردن جواب میرویم و اگر جواب پیدا نشد آن استیت را از بازدید شده ها حذف میکنم.

به عنوان مثال(عمق ۲) ریشه همه ی بچه های خود را بازدید میکند و سپس بچه سمت چپی ریشه همه بچه های خود را از سمت چپ بازدید میکند و...

۳) تخمین ها: فاصله منهتنی: تعداد حرکاتی که هر عامل هوشمند باید طی کند تا به هر غذایی که برایش قابل خوردن است برسد. به این صورت که مثلا مختصات p منهای مختصات همه غذاهایی که میتواند بخورد و هر کدام که کمتر شد h(n) آن استیت میشود. و f(n) آن میشود عمقی که تا الان این عامل آمده است + کمترین هزینه تا غذای بعدی(delta x + delta y) و سپس این وضعیت در صورت تکراری نبودن (مکان های عامل ها و تعداد غذاها و پوزیشن غذاها) وارد صف شده و از صف کمترین f(n) را برداشته و گسترش میدم.

Admissible:

در راه رسیدن به غذا هم دیوار ممکن است وجود داشته باشد و هم عامل دیگر و هم غذای زهردار. دراین صورت عامل هوشمند باید مسیری که زودتر از همه آن را به غذا میرساند را عوض کند تا به این عوامل

برخورد نکند. اگر هم که درمسیرش مانعی نباشد با همین تعداد قدم محاسبه شده به مسیر میرسد. پس در هر صورت $h(n) \le h^*(n)$.

۴) تفاوت ها و مزیت ها:

از آنجایی که الگوریتم BFS و IDS به شکل ناآگاهانه است, به مراتب زمان بسیار بیشتری نسبت به الگوریتم A* طول میکشند. چرا که بدون هیچ سیاست و هوشمندی, کل فضا را جستجو میکنند. اما A* با یک روشی هوشمندانه تر درجستجو شرکت میکند. هزینه زمانی الگوریتم های BFS و IDS از جنس (O(b^d است که در آن b حداکثر تعداد شاخه ها و m طول بیشترین مسیرممکن در گراف است. اما در A* هزینه ی زمانی الگوریتم برابر است با تعداد نودهایی که در آن ها g(n) + h(n) کوچکتر مساوی جواب اصلی است. BFS نیاز به حافظه بیشتری نسبت به IDS دارد چرا که هزینه‌ی حافظه‌ای BFS از جنس O(b^d) است درحالی که هزینه‌ی حافظه‌ای O(b*d) است. هزینه حافظه ای A* نیز مانند هزینه زمانی الگوریتم آن است. باید توجه داشت که هر سه الگوریتم کامل (A* در صورتی که Admissable باشد) میباشند و در اینجا هر سه جواب optimal را میدهند.

5)

زمان اجرا	تعدا استیت های مجزای دیده شده	تعداد استیت های دیده شده	فاصله جواب	نقشه ۱
۵۶,۴۷۳ ثانیه	۱۴۳۱۰۲	۹۱۴۷۹۸	۳۳	BFS
۵۳ دقیقه	۱۵۹۸۶۶۶۵	۵۴۶۸۴۴۱۴	۳۳	IDS
۵۴۵,۲۴ ثانیه	۱۱۳۹۵۶۵	۵۸۹۵۹۰۷	۳۴	A*

زمان اجرا	تعدا استیت های مجزای دیده شده	تعداد استیت های دیده شده	فاصله جواب	نقشه ۲
۱,۱۱ ثانیه	۳۵۴۹	۲۰۹۴۵	۱۷	BFS
۱۹,۱ ثانیه	۳۷۳۵۸۰	۱۵۲۴۰۸	۱۷	IDS
۴,۱۳ ثانیه	۱۲۹۵۱	۵۵۳۳۹	۱۷	A*

زمان اجرا	تعدا استیت های مجزای دیده شده	تعداد استیت های دیده شده	فاصله جواب	نقشه ۳
۰,۸۵ ثانیه	۲۶۸۷	۱۵۱۱۶	۲۰	BFS
۲۴,۵۵ ثانیه	۱۶۸۶۹۳	۴۷۳۴۳۳	۲۰	IDS
۳,۱۲ ثانیه	۱۱۸۷۱	۴۸۱۹۰	۲۰	A*

زمان اجرا	تعدا استیت های مجزای دیده شده	تعداد استیت های دیده شده	فاصله جواب	نقشه ۴
۳,۴۱ ثانیه	۱۰۶۲۲	۶۴۱۸۸	۱۷	BFS
۵۴,۰۲ ثانیه	۴۲۵۳۱۷	۱۰۲۲۵۳۹	۱۷	IDS
۴,۲ ثانیه	۵۱۴۶۹	۱۴۷۰۸	۱۷	A*

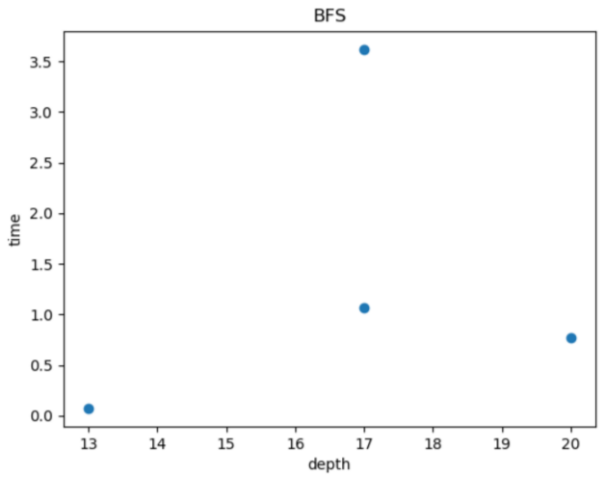
زمان اجرا	تعدا استیت های مجزای دیده شده	تعداد استیت های دیده شده	فاصله جواب	نقشه ۵
۰,۰۹۱ ثانیه	۳۳۵	۱۵۰۸	۱۳	BFS
۱,۲۱ ثانیه	۱۱۵۲۰	۲۷۷۵۷	۱۳	IDS
0.45 ثانیه	۱۱۶۱	۴۴۴۷	۱۳	A*

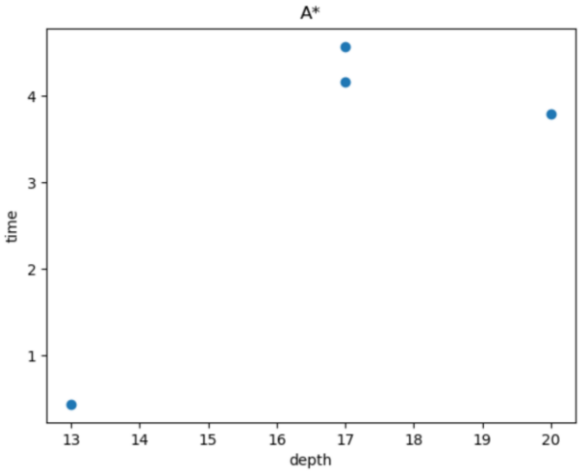
۵) در جدول بالا:

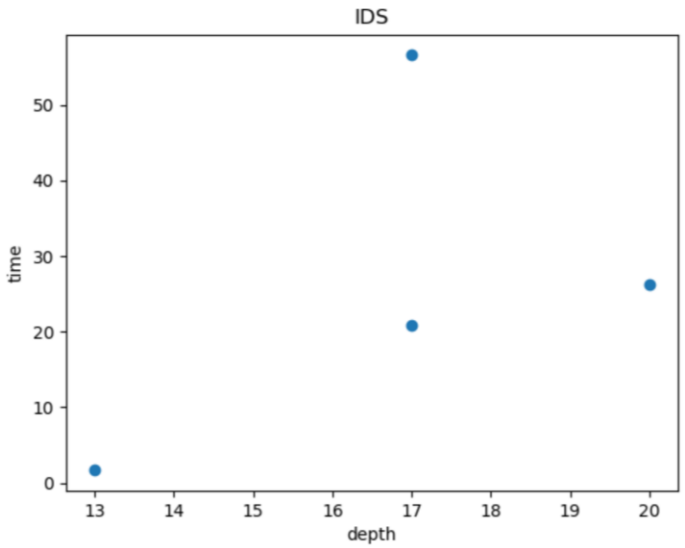
فاصله جواب = تعداد حرکتهای انجام شده برای رسیدن به پاسخ درست

تعداد استیت های دیده شده = تعداد حرکات انجام شده درالگوریتم برای رسیدن به جواب.

۶) نمودار ها برای تست های ۲و ۳و ۴ و۵


BFS
time
depth
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
13
14
15
16
17
18
19
20


A*
time
depth
1
2
3
4
13
14
15
16
17
18
19
20


IDS
time
depth
0
10
20
30
40
50
13
14
15
16
17
18
19
20