

سوال: آیا ترتیب کاوش همان ترتیبی بود که انتظار داشتید؟ آیا پکمن در راه رسیدن به هدف، به همه مربع‌های کاوش شده می‌رود؟

بله، در این الگوریتم همانطور که انتظار داشتیم در ابتدا عمق را جستجو میکنیم همانطور که از الگوریتم dfs انتظار داشتیم.

خیر با اجرا کردن این الگوریتم مشاهده میکنیم که این الگوریتم برخی از مربع‌ها را نمی‌پوشاند.

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

خیر، زیرا در این الگوریتم هر مسیر را تا انتها کاوش میکنیم و اگر در وسط مسیر راه بهینه‌تری برای پیمایش داشته باشیم این الگوریتم ابتدا مسیری که انتخاب کرده را تا انتها پیمایش میکند که این برای ما مطلوب نیست.

آیا الگوریتم جستجوی اول سطح راه‌حل با کمترین هزینه را پیدا می‌کند؟ اگر نه پیاده‌سازی خود را چک کنید.

بله در این الگوریتم جستجو ما همیشه کمترین هزینه را پیدا میکنیم

سوال: الگوریتم‌های جستجویی که تا به این مرحله پیاده‌سازی کرده اید را روی openMaze اجرا کنید و توضیح دهید چه اتفاقی می‌افتد.

هنگامی که الگوریتم‌های مختلف جستجو را روی openMaze اجرا میکنیم مشاهده میکنیم که هزینه‌های جستجو‌ها برای dfs, ucs, aStar با هم دیگر برابر هستند اما هزینه bfs برای اجرا چندین برابر بیشتر است اما این رابطه برای node های بسط داده شده برعکس است و الگوریتم node, bfs های بیشتری را به نسبت ۳ الگوریتم دیگر بسط میدهد.

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

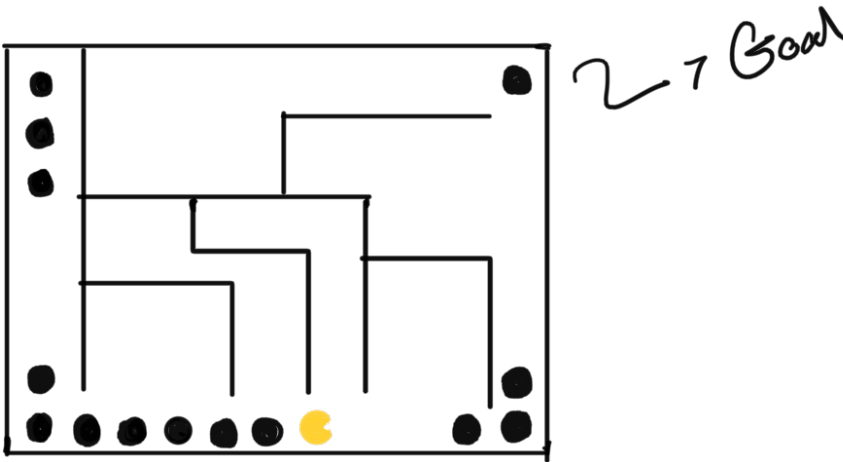
در این قسمت با استفاده از manhattanDistance فاصله را تا هر ۴ گوشه پیدا میکنیم و به سمت نزدیکترین حرکت میکنیم و این روند را تا زمانی که آخرین گوشه را بخوریم ادامه میدهیم همچنین این هیوریستیک consistent هم هست به دلیل اینکه فاصله منتهی همواره از فاصله واقعی کمتر است.

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

در این قسمت با استفاده از mazeDistance فاصله pacman تا غذا‌ها را به دست می‌آوریم و در اینجا بیشترین فاصله تا غذا را برای تابع هیوریستیک در نظر میگیریم بنابراین این هیوریستیک نیز consistent است.

سوال: `ClosestDotSearchAgent` شما، همیشه کوتاه ترین مسیر ممکن در ماز را پیدا نخواهد کرد. مطمئن شوید که دلیل آن را درک کرده اید و سعی کنید یک مثال کوچک بیاورید که در آن رفتن مکرر به نزدیکترین نقطه منجر به یافتن کوتاهترین مسیر برای خوردن تمام نقاط نمی شود.

در این الگوریتم ما همواره به سمت نزدیکترین نقطه میرویم که این عملکرد ممکن است برای ما در نهایت بهینه نباشد.



در اینجا مشاهده میکنیم که الگوریتم برای اینکه در هر مرحله به سراغ نزدیکترین نقطه می رود چندین مسیر تکراری اینجا تکرار میکند که این برای ما مطلوب نیست.