

1. سوال اول: ترتیب کاوش بر اساس ترتیب ورودی successors است و با تغییر فانکشن getSuccessors ممکن است مسیر یافت شده نیز تغییر کند. پکمن فقط مسیر یافت شده را طی میکند و مربع های کاوش شده ممکن است در مسیر یافت شده نباشند.

سوال دوم: خیر، پیدا کردن successors به ترتیب زیر است:

Directions.NORTH, Directions.SOUTH, Directions.EAST,
Directions.WEST

لذا این روش نمیداند کدام انتخاب جواب بهینه دارد و ممکن است جواب یافت شده بهینه نباشد.

4. در الگوریتم A^* ، بر اساس فانکشن heuristic، انتخاب حرکت مناسب بستگی به هزینه ای دارد که فانکشن heuristic برمیگرداند. در این الگوریتم در هر مرحله از بررسی گره ها، گره ای که کمترین هزینه را دارد در اولویت بالاتری برای بررسی قرار میگیرد.

6. در این سوال، با یک الگوریتم ساده، در ابتدا، نزدیکترین گوشه را پیدا میکنیم، هزینه ی انتقال به این گوشه برابر با فاصله ی این انتقال است که از تابع manhattanDistance به دست آورده می شود. سپس با فرض انجام انتقال فوق، دوباره از میان گوشه های باقی مانده، نزدیک ترین گوشه را پیدا کرده و انتقال را انجام میدهیم. مقدار خروجی این تابع جمع هزینه های این انتقال هاست. این مقدار از هزینه واقعی رسیدن به مقصد کمتر است. در مرود سازگار بودن نیز با انجام هر حرکت با توجه به این که میزان هیورستیک جمع حرکت ها تا رسیدن به همه اهداف است، لذا مقداری که از هیورستیک کم میشود همواره کوچکتر و یا اگر حرکت فقط یک مرحله باشد برابر با مقدار هیورستیک است

7. در این سوال، مقدار بازگردانده شده برابر با ماکزیم فاصله پک من از غذا است.

8. چون این الگوریتم در هر مرحله نزدیکترین غذا را پیدا میکند و ممکن است بهترین جواب، در برخی مرحله ها لزوما نزدیکترین نباشد. مثلا هزینه حرکت از نقطه 1 به 2 کمتر از هزینه 1 به 3 باشد ولی در مجموع هزینه 1 به 2 و سپس به 3، بیشتر از هزینه 1 به 3 سپس به 2 باشد