

بخش ۲:

سوال: چرا باید تابع **normalize** را در این بخش فراخوانی کنیم؟ لزوم استفاده از آن را شرح دهید.

```

*** YOUR CODE HERE ***
for gP in self.allPositions:
    self.beliefs[gP] = self.getObservationProb(observation,
        gameState.getPacmanPosition(), gP, self.getJailPosition()) *
    self.beliefs[gP]
self.beliefs.normalize()

```

چون احتمال وجود روح در هر خانه را در احتمال جدیدی که بدست آورده ایم میکنیم، پس هر احتمال هر خانه کمتر یا مساوی احتمال قبلی خود خواهد بود. این باعث میشود جمع احتمال ها از یک کمتر باشد در حالی که میدانیم روح حتما در یکی از این خانه ها قرار دارد. پس نورمالایز میکنیم تا جمع احتمال ها ۱ شود.

سوال: توضیح دهید مقدار **beliefs** در گذر زمان چگونه تغییر میکند؟

دقیق تر میشود چرا که خانه هایی که طبق مشاهدات جدید، احتمال کمتری برای وجود روح دارند، در عدد کوچک تری ضرب میشوند. همچنین احتمال ها متمرکز تر میشوند به طوری که احتمال وجود روح در ابتدا در همه ی خانه ها یکسان است اما پس از مدتی، این احتمالات اختلاف پیدا میکنند و به جواب نزدیک تر میشوند.

بخش ۳:

سوال: کاربرد کلاس **DiscreteDistribution** را به همراه متدهای آن توضیح دهید.

این کلاس شامل یک دیکشنری است که **position: possibility** است. به صورتی که احتمال وجود روح در هر خانه را نگهداری میکند.

Copy یک دیپ کپی از احتمالات کنونی میسازد.

Total همه ی ولیو ها (احتمالات) را با هم جمع میکند و برمیگرداند.

Argmax کلید(لوکیشن) مربوط به بزرگترین ولیو(احتمال) را برمیگرداند.

Normalize همه ی ولیو(احتمال) ها را در نسبتی ضرب میکند که جمعشان برابر با ۱ شود.

Sample یک کلید از بین کلید ها انتخاب میکند. احتمالی که کدام انتخاب شود، با ولیو(احتمال) آن همنسبت است.

سوال: چگونه ساختار شبکه بیزین در پروژه وابستگیهای بین متغیرها را نشان میدهد؟

ما با دانستن موقعیت پکمن و noisy distance و همچنین وجود روح ها در زندان، میخواهیم به موقعیت واقعی روح هایی که دستگیر نشده اند برسیم. همچنین میدانیم که noisy distance با چه احتمالاتی حرکت میکند پس:

$P(\text{noisy distance} \mid \text{true distance})$

$P(\text{new ghost position} \mid \text{previous ghost position})$

مواردی هستند که به ما کمک میکنند تا $P(\text{true distance})$ را بدست آوریم.