

سوال 1 : چون بازی یک عامل کامپیوتر دارد و یک عامل طبیعی برای کامپیوتر از ماکس کننده و برای عامل طبیعی باید از شانس استفاده کنیم بنابراین از این الگوریتم ها استفاده میشود .

سوال 2 : زیرا اگر در واقعیت نمیتوانیم همه حالت ها را تا ترمینال ها ادامه دهیم چون بسیار بسیار زمان و فضا میخواد بنابراین استفاده از این دو الگوریتم کمک میکند تا تا یک عمقی بررسی شود و تا ترمینال ها یک تخمینی زده شود برای همین لزوما بهینه نیست جواب ما.

گزارش :

expectimax : 1 در این الگوریتم از دو تابع **chance** و **maximizer** استفاده میشود که تابع اول برای عامل هوشمند هست و تابع دوم برای عامل محیطی هست در اولی بین فرزندان **max** میگیریم و در تابع دوم بین فرزندان باید میانگین وزن دار بگیریم.

در تابع **evaluation** برای عامل یکنوایی به خانه بالا سمت چپ بیشترین ارزش و به بقیه عضو های در امتداد آن به ترتیب نزولی امتیاز میدهیم و در عضو های اصلی **board** ضرب میکنیم . برای عامل دوم برای به حداقل رساندن اختلاف دو خانه مجاور اختلاف آن ها را از امتیاز کل با یک ضربی کم میکنیم و با امتیاز عامل اول جمع میکنیم.

mcts : 2 در این بخش به تعداد برابر اکشن ها را تست میکنیم و بقیه بازی را با انجام فرضی این اکشن انجام میدهیم و از مجموع امتیازات میانگین میگیریم و در آخر اکشنی با بیشترین امتیاز را انتخاب میکنیم

ucb : 3 این الگوریتم ورژن 2 الگوریتم بالا هست و تفاوت این است که به صورت برابر تست نمیکنیم بلکه با استفاده از دو عامل **promising** و **uncertain** نسبت تعداد تست را تعیین میکنیم که در تابع **ucb** هندل میشود