Ali Padyav – 810199388 CA2 – Game

بدون هرس آلفا و بتا

عمق ۱:

```
PS D:\Uni\AI\3\CA2\Game> py main.py 1 0
0.05233454704284668
{'red': 72, 'blue': 28}
72.0
PS D:\Uni\AI\3\CA2\Game> py main.py 1 0
0.07078361511230469
{'red': 74, 'blue': 26}
74.0
PS D:\Uni\AI\3\CA2\Game> py main.py 1 0
0.05422496795654297
{'red': 70, 'blue': 30}
70.0
```

عمق ۳:

```
PS D:\Uni\AI\3\CA2\Game> py main.py 3 0
2.902399778366089
{'red': 71, 'blue': 29}
71.0
PS D:\Uni\AI\3\CA2\Game> py main.py 3 0
2.908679246902466
{'red': 71, 'blue': 29}
71.0
PS D:\Uni\AI\3\CA2\Game> py main.py 3 0
2.9494988918304443
{'red': 78, 'blue': 22}
78.0
```

عمق ۵:

```
PS D:\Uni\AI\3\CA2\Game> py main.py 5 0
319.83496499061584
{'red': 82, 'blue': 18}
82.0
```

چون زمان زیادی گرفت یکبار آزمایش شد

با هرس آلفا و بتا

عمق ۱:

```
PS D:\Uni\AI\3\CA2\Game> py main.py 1 0
0.05513167381286621
{'red': 64, 'blue': 36}
64.0
PS D:\Uni\AI\3\CA2\Game> py main.py 1 0
0.05304455757141113
{'red': 69, 'blue': 31}
69.0
PS D:\Uni\AI\3\CA2\Game> py main.py 1 0
0.05407094955444336
{'red': 68, 'blue': 32}
68.0
```

عمق ۳:

```
PS D:\Uni\AI\3\CA2\Game> py main.py 3 0
0.6149909496307373
{'red': 77, 'blue': 23}
77.0
PS D:\Uni\AI\3\CA2\Game> py main.py 3 0
0.6499052047729492
{'red': 75, 'blue': 25}
75.0
PS D:\Uni\AI\3\CA2\Game> py main.py 3 0
0.6472549438476562
{'red': 64, 'blue': 36}
64.0
```

عمق ۵:

```
PS D:\Uni\AI\3\CA2\Game> py main.py 5 0
10.056483268737793
{'red': 81, 'blue': 19}
81.0
PS D:\Uni\AI\3\CA2\Game> py main.py 5 0
9.800293207168579
{'red': 76, 'blue': 24}
76.0
PS D:\Uni\AI\3\CA2\Game> py main.py 5 0
10.062143564224243
{'red': 83, 'blue': 17}
83.0
```

عمق ۷:

```
PS D:\Uni\AI\3\CA2\Game> py main.py 7 0
109.65971565246582
{'red': 79, 'blue': 21}
79.0
```

واضح است هرچه عمق بیشتر میشود شانس پیروزی ما هم بیشتر میشود. همینطور شانس پیروزی با هرس و بدون هرس با عمق یکسان تفاوت آنچنانی نمیکند. ولی زمان بدون هرس ملموسا بیشتر است به طوری که نتوانستم با عمق ۷ نتیجه ای به دست آورم.

سوالات

۱- هیوریستیک یا evaluator خوب باید بتواند عمق های بالاتر پیشبینی اش به سود ما نزدیکتر باشد. همینطور نباید عددی خیلی کوچک باشد. زیرا بین استیت های مختلف تمایز ایجاد نمیکند و انتخاب ما تا حدی شانسی میشود.

۲-هرچه عمق الگوریتم بیشتر باشد محاسبات ما به برگ ها نزدیک تر است. به همین دلیل گره های بیشتری را ویزیت میکنیم و شانس پیروزی ما هم بیشتر است. همینطور باعث افزایش زمان به طور قابل توجهی میشود (نمایی).

۳- برای هرس کردن اگر در تابع min هستیم بهتر است فرزندان را از کوچک به بزرگ ببینیم چون در این صورت احتمال اینکه آن فرزند از آلفا کوچکتر باشد هست و زودتر هرس انجام میشود. به همین دلیل وقتی در تابع max هستیم برعکس بهتر است اول فرزندی که بزرگتر است را ببینیم.