سوال حل مکعب روبیک

برای حل این سوال با توجه به اینکه رفتار کاملا حریصانه برای همرنگ کردن بیشترین مهره های ممکن با مرکز هر وجه الزاما منجر به حل مکعب نمیشود و خیلی اوقات بهتر است حرکاتی را انجام دهیم که ترتیب خاصی که در مکعب ایجاد شده را بهم بزند تا در آینده بتوانیم به الگویی در ترتیب مهره ها برسیم که مکعب را قابل حل کند از الگوریتمSimulated Annealingبرای حل آن استفاده کرده ام که رفتار کاملا حریصانه ندارد
همسایه های هر استیت شامل حالاتی اند که با یکی از ۱۲ حرکت مجاز روی مکعب به وجود می آیند بنابراین در الگوریتم عدد تصادفی بین ۱ تا ۱۲ انتخاب شده و حرکت متناظر آن عدد انجام می شود
برای چرخش ساعتگرد هر وجه ابتدا آن را در ماتریس با سطر اول برابر ۱و۰و۰ و سطر دوم ۰و۱و۰ و سطر سوم ۰و۰و۱ ضرب کرده و سپس آن را ترانهاده می کنیم. برای چرخش پاد ساعتگرد ابتدا ترانهاده و سپس در ماتریس ضرب میکنیم علاوه بر این وقتی یک وجه را میچرخانیم ردیف هایی از وجوه همسایه آن نیز چرخش می یابند که در کد مشخص شده است
سپس همسایه تصافی انتخاب شده توسط تابع هدف ارزیابی می شود. تابع هدف به این صورت تعریف شده است که در هر وجه تعداد مربع هایی که رنگ آنها متفاوت از مرکز وجه است و سر جای مناسب نیستند را می شمارد مکعب زمانی حل میشود که مقدار تابع هدف برای آن صفر شود البته تابع های هدف مناسب تری هم میتوان در نظر گرفت مثلا تعداد مهره های لبه یا گوشه نا هماهنگ و ترکیب مناسبی از این عوامل با ضریب مناسب یا تابع هدف را بر اساس ایجاد الگوهایی در مهره ها که می توانند طی چند حرکت منجر به حل مکعب شوند در نظر بگیریم که دشوار تر است به هر حال تابع هدف کنونی در الگوریتم الزاما تابع هدف مناسبی نیست چون حریصانه است و ممکن است همسایه ای که تعداد مهره های نا همخوان بیشتری نسبت به همسایه دیگری دارد شامل الگوهایی از مهره باشد که بهتر و سریعتر منجر به حل مکعب شود

برای گرفتن نتایج بهتر مقدار اولیه دما و نرخ کاهش دما باید تیون شوند و مقدار کنونی بهینه نبوده و فقط گاهی اوقات منجر به حل مکعب میشوند

```
        f.write(f"\n")

        for i in range(3):
            for j in range(3):
                f.write(f"{right[i][j]} ")
            f.write(f"\n")
        f.write(f"\n")

        for i in range(3):
            for j in range(3):
                f.write(f"{front[i][j]} ")
            f.write(f"\n")
        f.write(f"\n")

        for i in range(3):
            for j in range(3):
                f.write(f"{down[i][j]} ")
            f.write(f"\n")
        f.write(f"\n")

        for i in moves:
            f.write(f"{i}\n")


    print("The value of objective function is (it must be 0)",best_solution_value)
```

```
78]  ✓ 0.0s
··  The value of objective function is (it must be 0) 0
```