بسمه تعالی

دانشکده مهندسی برق و کامپیوتر
دانشگاه صنعتی اصفهان
یادگیری ماشین- نیمسال اوّل ۱۴۰۴ - ۱۴۰۳
تکلیف شماره ۲ – تحویل شنبه ۱۴۰۳/۹/ ۳

سپهر عبادی

۹۹۳۳۲۴۳

۱.

- مقدار بیشینه جهانی تابع sinc برابر ۱ است.

به دلیل اینکه طبق گفته مسئله مخرج کسر زمانی که صفر شود تابع مقدار ۱ را برمی گرداند. پس حد این تابع به صورت زیر است :

$$\lim_{x,y\to 0} \frac{x^2+y^2}{\sqrt{x^2+y^2}} = 1$$

پس برای هر مقدار از دیگر از x,y خروجی تابع کمتر از ۱ خواهد بود، زیرا داریم :

$$|\sin(x^2+y^2)| \le 1$$

- در تابع sinc_gradian مشتقات جزئی تابع محاسبه شده است. مشتق جزئی تابع به صورت زیر محاسبه می شود:

$$\left(\frac{\sin(x^2+y^2)}{\sqrt{x^2+y^2}}\right)\frac{\partial}{y\partial} = \frac{sinc\partial}{y\partial}, \left(\frac{\sin(x^2+y^2)}{\sqrt{x^2+y^2}}\right)\frac{\partial}{x\partial} = \frac{sinc\partial}{x\partial}$$

مشتق کسری با استفاده از فرمول زیر و به صورت زیر بدست می آید:

$$\frac{f'(x)g(x) - f(x)g'(x)}{^2[g(x)]} = \left(\frac{f(x)}{g(x)}\right)\frac{d}{dx}$$

```
PS E:\IUT\Lessons\Semester 9th\AI\HW\HW2> python -u "e:\IUT\Lessons\Semester 9th\AI\HW\HW2\.venv\Q1.py"
Results :
1: Max in --> (5.5402, -5.2361) with the value --> 0.1312.
2: Max in --> (6.4451, 9.2569) with the value --> 0.0887.
3: Max in --> (9.4165, -0.9275) with the value --> 0.1057.
4: Max in --> (2.1130, 5.3392) with the value --> 0.1741.
5: Max in --> (2.7780, 4.3552) with the value --> 0.1935.
6: Max in --> (-9.2708, -4.0187) with the value --> 0.0990.
7: Max in --> (-8.7703, 7.0931) with the value --> 0.0887.
8: Max in --> (-2.6071, 3.6877) with the value --> 0.2214.
9: Max in --> (-4.8684, -3.0446) with the value --> 0.1741.
10: Max in --> (-9.7077, -2.8033) with the value --> 0.0990.
❌ Not Reach to global Max.
```

۲.

حالات اولیه : چون در فضای باور هستیم پس حالت اولیه شامل همه حالت ها است.

اعمال : جارو کردن، رفتن از یک خانه به خانه دیگر

هر گره یک حالت باور است.

هدف : تمیز شدن همه خانه ها

در کد باید ابتدا فضای باور را بسازیم. که شامل تمام خانه ها میشود و همچنین خانه هایی که کثیف هستند.

سپس حالت هدف را مشخص باید کنیم. که شامل همه خانه ها به صورت تمیز است.

در هر گام از الگوریتم از وضعیت جاری و دنباله عملیات ها از سر صف برداشته میشوند وبررسی می شوند اگر حالت هدف بودند که دنباله اعمال را بر می گرداند در غیر این صورت باید دنباله اعمال را بسازد.

برای هر موقعیت ربات، اگر آن موقعیت کثیف باشد (در مجموعه current_dirt باشد)، خانه تمیز می‌شود. وضعیت جدید با خانه جاروشده به روز می‌شود و اگر این وضعیت قبلاً بررسی نشده باشد، در صف قرار می‌گیرد.

برای هر موقعیت ربات، همسایگان آن (موقعیت‌های اطراف) تولید می‌شود. وضعیت جدید با موقعیت‌های جدید ربات به روز می‌شود و اگر این وضعیت قبلاً بررسی نشده باشد، در صف قرار می‌گیرد.

```
PS E:\IUT\Lessons\Semester 9th\AI\HW\HW2> python -u "e:\IUT\Lessons\Semester 9th\AI\HW\HW2\.venv\Q2.py"
X - X
- X -
X - X
Actions:
('clean', (0, 0))
('move', (0, 0), (1, 0))
('move', (1, 0), (2, 0))
('clean', (2, 0))
('move', (2, 0), (1, 0))
('move', (1, 0), (1, 1))
('clean', (1, 1))
('move', (1, 1), (0, 1))
('move', (0, 1), (0, 2))
('clean', (0, 2))
('move', (0, 2), (1, 2))
('move', (1, 2), (2, 2))
('clean', (2, 2))
PS E:\IUT\Lessons\Semester 9th\AI\HW\HW2>
```

```
PS E:\IUT\Lessons\Semester 9th\AI\HW\HW2> python -u "e:\IUT\Lessons\Semester 9th\AI\HW\HW2\.venv\Q2.py"
X X -
X - -
X - X
Actions:
('clean', (0, 0))
('move', (0, 0), (0, 1))
('clean', (0, 1))
('move', (0, 1), (1, 1))
('move', (1, 1), (1, 0))
('clean', (1, 0))
('move', (1, 0), (2, 0))
('clean', (2, 0))
('move', (2, 0), (2, 1))
('move', (2, 1), (2, 2))
('clean', (2, 2))
```

۳.

سوال یک را با استفاده از الگوریتم سرد شدن شبیه سازی شده پیاده سازی کردیم که این الگوریتم تفاوت هایش با استفاده از گرادیان و الگوریتم عرض نخست در سوال یک به صورت زیر است :

این روش با این که کند تر است اما چون با یک احتمالی از نقاط تصادفی شروع میکند و همچنین چون با یک احتمالی نقاط بد را هم در نظر میگیرد، احتمال اینکه در بهینه محلی گیر کند کاهش یافته و احتمال اینکه به بهینه جهانی برسد بیشتر از روش گرادیان است.

در این روش اگر به خروجی ها دقت کنید مشاهده می شود که میانگین مقادیر بزرگ تر از میانگین مقادیر در روش قبلی است که از این می توان نتیجه گرفت احتمال رسیدن به بهینه جهانی بیش تر است.

```
PS E:\IUT\Lessons\Semester 9th\AI\HW\HW2> python -u "e:\IUT\Lessons\Semester 9th\AI\HW\HW2\.venv\Q3.py"
Results :
1: Max in --> (2.9371, -3.4852) with the value --> 0.2059.
2: Max in --> (-7.8439, -9.5195) with the value --> 0.0791.
3: Max in --> (0.0173, 1.2019) with the value --> 0.8253.
4: Max in --> (2.7873, -2.5470) with the value --> 0.2630.
5: Max in --> (-4.8626, -1.8635) with the value --> 0.1759.
6: Max in --> (3.1734, -3.2343) with the value --> 0.2193.
7: Max in --> (0.4615, -0.6971) with the value --> 0.7696.
8: Max in --> (0.1990, -1.0313) with the value --> 0.8499.
9: Max in --> (-1.0166, 3.5929) with the value --> 0.2628.
10: Max in --> (0.4939, -0.8128) with the value --> 0.8266.
❌ Not Reach to global Max.
PS E:\IUT\Lessons\Semester 9th\AI\HW\HW2>
```

```
PS E:\IUT\Lessons\Semester 9th\AI\HW\HW2> python -u "e:\IUT\Lessons\Semester 9th\AI\HW\HW2\.venv\Q3.py"
Results :
1: Max in --> (-0.4042, 1.0003) with the value --> 0.8512.
2: Max in --> (-0.9148, -3.6190) with the value --> 0.2624.
3: Max in --> (5.2534, 2.2872) with the value --> 0.1724.
4: Max in --> (-11.9633, -4.6751) with the value --> 0.0778.
5: Max in --> (6.2342, -6.2048) with the value --> 0.1049.
6: Max in --> (-3.0465, -8.6053) with the value --> 0.1092.
7: Max in --> (1.0477, -0.2526) with the value --> 0.8512.
8: Max in --> (-6.2809, 5.5600) with the value --> 0.1131.
9: Max in --> (7.8724, 2.8974) with the value --> 0.1133.
10: Max in --> (7.4529, 5.8386) with the value --> 0.1051.
❌ Not Reach to global Max.
```