

تمرین سری دوم درس هوش مصنوعی دکتر ساجدی

تاریخ تحویل: یکشنبه ۱۳ اردیبهشت

تاریخ تحویل حضوری: دوشنبه ۱۴ اردیبهشت

با استفاده از الگوریتم Simulated Annealing مساله فروشنده دوره‌گرد را برای دیتاست bayg29 که شامل ۲۹ شهر است پیاده‌سازی نمایید.

قسمت نمره اضافی: نمودار تغییرات انرژی و دما در طی اجرای برنامه را رسم نمایید.

راهنمایی: simulated annealing را در wikipedia مطالعه نمایید زیرا چرخه اجرایی مانند این سایت نوشته شده است.

مساله فروشنده دوره‌گرد:

(به انگلیسی: Travelling salesman problem به اختصار: TSP) مسئله‌ای مشهور است که ابتدا در سده ۱۸ مسائل مربوط به آن توسط ویلیام همیلتون و توماس کرکمن مطرح شد و سپس در دهه ۱۹۳۰ شکل عمومی آن به وسیله ریاضیدانانی مثل کارل منگر از دانشگاه هاروارد و هاسلر ویتنی از دانشگاه پرینستون مورد مطالعه قرار گرفت.

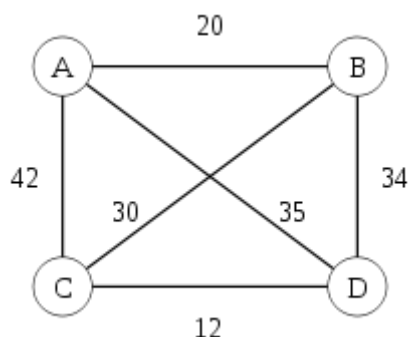
شرح مسئله بدین شکل است:

تعدادی شهر داریم و هزینه رفتن مستقیم از یکی به دیگری را می‌دانیم. مطلوب است کم‌هزینه‌ترین مسیری که از یک شهر شروع شود و از تمامی شهرها دقیقاً یکبار عبور کند و به شهر شروع بازگردد.

تعداد جواب‌های شدنی مسئله، برابر است با  $\frac{1}{2}(n-1)!$  برای  $n > 2$  که  $n$  تعداد شهرها می

باشد. در واقع این عدد برابر است با تعداد دورهای همیلتونی در یک گراف کامل با  $n$  رأس.

برای مثال شکل زیر یک گراف که این مساله را برای چهار شهر مدل می‌کند نمایش می‌دهد. روی هر یال هزینه رفتن از دو شهر که رأس‌های آن یال هستند نوشته شده است.



وظیفه شما تکمیل فایل‌های با پسوند m. زیر می‌باشد.

**تذکر:** می‌توانید خودتان فایل‌هایی بسازید و برنامه را به نحوی دیگر پیاده‌سازی نمایید. قرار دادن این فایل‌ها برای کمک به ساختار دادن برنامه می‌باشد، همچنین می‌توانید ورودی و خروجی‌های هر تابع را هر قدر که می‌خواهید تغییر دهید.

#### simulatedAnnealing.m

این فایل تقریباً کامل است البته برخی قسمت‌های آن کامنت دارند که باید توسط شما تکمیل گردند.

این فایل چرخه اصلی برنامه است و پس از پایان تمامی فایل‌ها این فایل باید جواب مساله را به شما بدهد.

#### readFile.m

این فایل باید دیتابیس را بخواند و یک ماتریس بسازد توضیحات مربوط به کاری که باید در این فایل را انجام دهید در پایان همین راهنما در قسمت [توضیحات در مورد دیتابیس](#) مطالعه نمایید. ورودی آن fileAddress می‌باشد که آدرس فایلی است که باید باز شود و خوانده شود و خروجی از روی آن ساخته شود.

خروجی آن data می‌باشد که ماتریسی است که در قسمت [توضیحات در مورد دیتابیس](#) گفته شده است باید به چه شکلی باشد.

#### initialState.m

این فایل یک جواب اولیه از مساله مورد نظر را پیدا می‌کند.

این تابع ورودی ندارد و خروجی آن generatedState می‌باشد که یک جواب برای مساله است. (انتخاب هوشمندانه این جواب نیز می‌تواند در الگوریتم تأثیر بگذارد ولی می‌توان از یک جواب رندم هم استفاده نمود)

#### scheduleTemp.m

این فایل مربوط به زیرتابعی است که برنامه‌ریزی دما را ارائه می‌دهد معمولاً باید خروجی این فایل

به گونه‌ای باشد که در ابتدای کار اعداد بزرگتری و در انتها اعداد کوچکتری بدهد. ورودی آن  $t$  می‌باشد که یک عدد است و خروجی آن `finalTemp` می‌باشد که با توجه به برنامه‌ریز تصمیم می‌گیرد چه عددی را برگرداند باید هر چه اجرای برنامه جلو می‌رود دما کمتر شود.

`getNeighbours.m`

این فایل `state`‌های همسایه یا یک راه حل جدید را با توجه به راه‌حلی که در آن قرار داریم ایجاد می‌نماید. الگوریتم این قسمت هر قدر هوشمندانه‌تر باشد احتمال رسیدن به جواب بهتر بیشتر است.

ورودی این تابع `state` می‌باشد که یک جواب از مساله است و خروجی آن `neighbourState` می‌باشد که جواب دیگری از مساله است و می‌توان با راه‌های مختلفی آن را ساخت به عنوان مثال می‌توانید جای دو شهر را به صورت رندم با یکدیگر عوض کنید.

`energy.m`

این فایل بررسی می‌کند که انرژی `state` ای که در آن هستیم چقدر هست چون در این مساله به دنبال مینیم کردن طول مسیر طی‌شده هستیم هر قدر جواب بهتر باشد باید انرژی کمتری داشته باشد. ورودی این یک `state` است و خروجی آن یک عدد می‌باشد.

`acceptanceProbability.m`

این فایل چک می‌کند که با توجه به دما و `state` قبلی و بعدی جواب جدید را قبول کنیم یا خیر. ورودی‌های این تابع `energy_s` که انرژی `state` ای می‌باشد که در آن هستیم، `energy_s_new` که انرژی `state` ای است که مربوط به همسایه است و  $t$  که دما می‌باشد هستند. این تابع چک می‌کند که آیا `state` جدید را باید انتخاب نماید یا خیر و این با توجه به دما اتفاق می‌افتد اگر دما کم باشد امکان انتخاب `state` جدید بدتر بیشتر می‌باشد و هر چه دما کم می‌شود احتمال `state` جدید بهتر بیشتر و `state` جدید بدتر کاهش یابد.

توضیحات در مورد دیتابیس:

محتویات فایل `bayg29` به این شکل می‌باشد.

NAME: bayg29

TYPE: TSP

COMMENT: 29 Cities in Bavaria, geographical distances (Groetschel,Juenger,Reinelt)

DIMENSION: 29

EDGE\_WEIGHT\_TYPE: EXPLICIT

EDGE\_WEIGHT\_FORMAT: UPPER\_ROW

DISPLAY\_DATA\_TYPE: TWOD\_DISPLAY

EDGE\_WEIGHT\_SECTION

97 205 139 86 60 220 65 111 115 227 95 82 225 168 103 266 205 149 120 58 257 152 52 180 136 82 34 145  
129 103 71 105 258 154 112 65 204 150 87 176 137 142 204 148 148 49 41 211 226 116 197 89 153 124 74  
219 125 175 386 269 134 184 313 201 215 267 248 271 274 236 272 160 151 300 350 239 322 78 276 220 60  
167 182 180 162 208 39 102 227 60 86 34 96 129 69 58 60 120 119 192 114 110 192 136 173 173  
51 296 150 42 131 268 88 131 245 201 175 275 218 202 119 50 281 238 131 244 51 166 95 69  
279 114 56 150 278 46 133 266 214 162 302 242 203 146 67 300 205 111 238 98 139 52 120  
178 328 206 147 308 172 203 165 121 251 216 122 231 249 209 111 169 72 338 144 237 331  
169 151 227 133 104 242 182 84 290 230 146 165 121 270 91 48 158 200 39 64 210  
172 309 68 169 286 242 208 315 259 240 160 90 322 260 160 281 57 192 107 90  
140 195 51 117 72 104 153 93 88 25 85 152 200 104 139 154 134 149 135  
320 146 64 68 143 106 88 81 159 219 63 216 187 88 293 191 258 272  
174 311 258 196 347 288 243 192 113 345 222 144 274 124 165 71 153  
144 86 57 189 128 71 71 82 176 150 56 114 168 83 115 160  
61 165 51 32 105 127 201 36 254 196 136 260 212 258 234  
106 110 56 49 91 153 91 197 136 94 225 151 201 205  
215 159 64 126 128 190 98 53 78 218 48 127 214  
61 155 157 235 47 305 243 186 282 261 300 252  
105 100 176 66 253 183 146 231 203 239 204  
113 152 127 150 106 52 235 112 179 221  
79 163 220 119 164 135 152 153 114  
236 201 90 195 90 127 84 91  
273 226 148 296 238 291 269  
112 130 286 74 155 291  
130 178 38 75 180  
281 120 205 270  
213 145 36  
94 217  
162

DISPLAY\_DATA\_SECTION

1 1150.0 1760.0  
2 630.0 1660.0  
3 40.0 2090.0  
4 750.0 1100.0  
5 750.0 2030.0  
6 1030.0 2070.0  
7 1650.0 650.0  
8 1490.0 1630.0  
9 790.0 2260.0  
10 710.0 1310.0  
11 840.0 550.0  
12 1170.0 2300.0  
13 970.0 1340.0  
14 510.0 700.0  
15 750.0 900.0  
16 1280.0 1200.0  
17 230.0 590.0  
18 460.0 860.0  
19 1040.0 950.0  
20 590.0 1390.0  
21 830.0 1770.0

22 490.0 500.0  
 23 1840.0 1240.0  
 24 1260.0 1500.0  
 25 1280.0 790.0  
 26 490.0 2130.0  
 27 1460.0 1420.0  
 28 1260.0 1910.0  
 29 360.0 1980.0

EOF

۷ خط اول فایل توضیحات در مورد فایل است که اصطلاحاً description نامیده می‌شود. پس از آن دو قسمت در فایل وجود دارد یکی EDGE\_WEIGHT\_SECTION که هزینه عبور از یک شهر به شهر دیگر است. خط اول ۲۸ عدد دارد که مربوط به هزینه حرکت از شهر اول به شهرهای دوم تا بیست و نهم می‌باشد. خط دوم ۲۷ عدد دارد که مربوط به هزینه حرکت از شهر دوم به شهرهای دیگر به جز شهر اول می‌باشد (زیرا هزینه حرکت از شهر اول به دوم را قبلاً داشته‌ایم) و همین‌طور الی آخر.

قسمت دوم DISPLAY\_DATA\_SECTION می‌باشد که مختصات جغرافیایی شهرها در آن نوشته شده است هر شهر یک x و یک y دارد که برای حالتی که بخواهیم گرافیکی این شهرها را نمایش دهیم استفاده می‌گردد یا گراف را رسم نماییم و در این تمرین ما کاری به آن نداریم. EOF به معنی end of file می‌باشد که در خط آخر نوشته شده است.

در قسمت readFile.m هدف این است که برنامه‌ای نوشته شود و اطلاعات این فایل را به صورت یک ماتریس ۲۹ در ۲۹ در آورد که هر درایه a و z آن هزینه حرکت از شهر a به شهر z می‌باشد. برای مثال دو سطر اول این ماتریس ۲۹ در ۲۹ در جدول زیر آورده شده است.

0	97	20	13	86	60	22	65	11	11	22	95	82	22	16	10	26	20	14	12	58	25	15	52	18	13	82	34	14
		5	9			0		1	5	7			5	8	3	6	5	9	0		7	2		0	6			5
97	0	12	10	71	10	25	15	11	65	20	15	87	17	13	14	20	14	14	49	41	21	22	11	19	89	15	12	74
		9	3		5	8	4	2		4	0		6	7	2	4	8	8			1	6	6	7		3	4	

موفق باشید