

بنام خدا

دانشکده‌ی مهندسی برق و کامپیوتر

درس تحقیق در عملیات

CA#1

شایان واصف

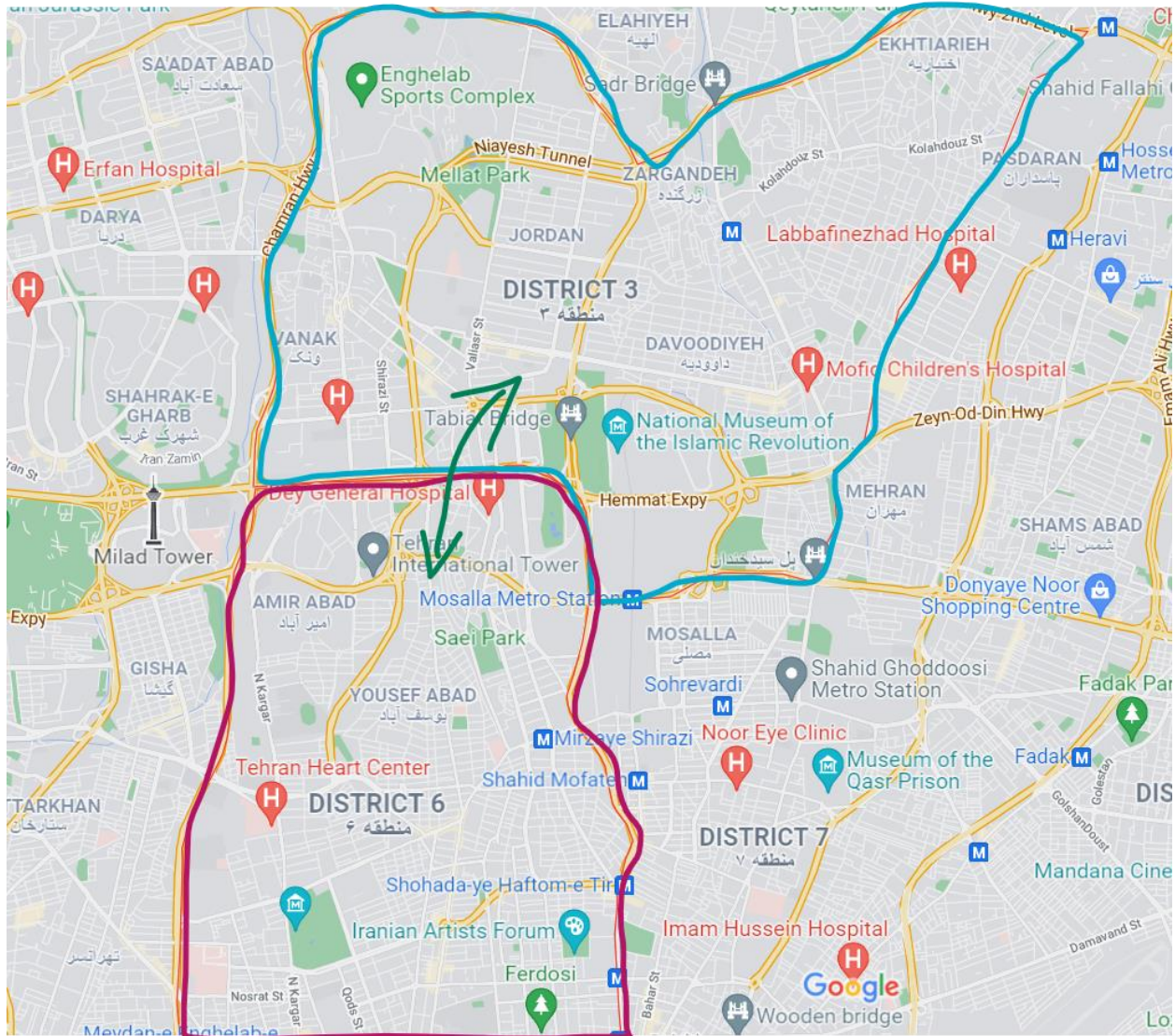
810197603

Table of Contents

Data Extraction:.....	2
Model Graph:	6
Minimum Cost Routing:.....	7

:Data Extraction

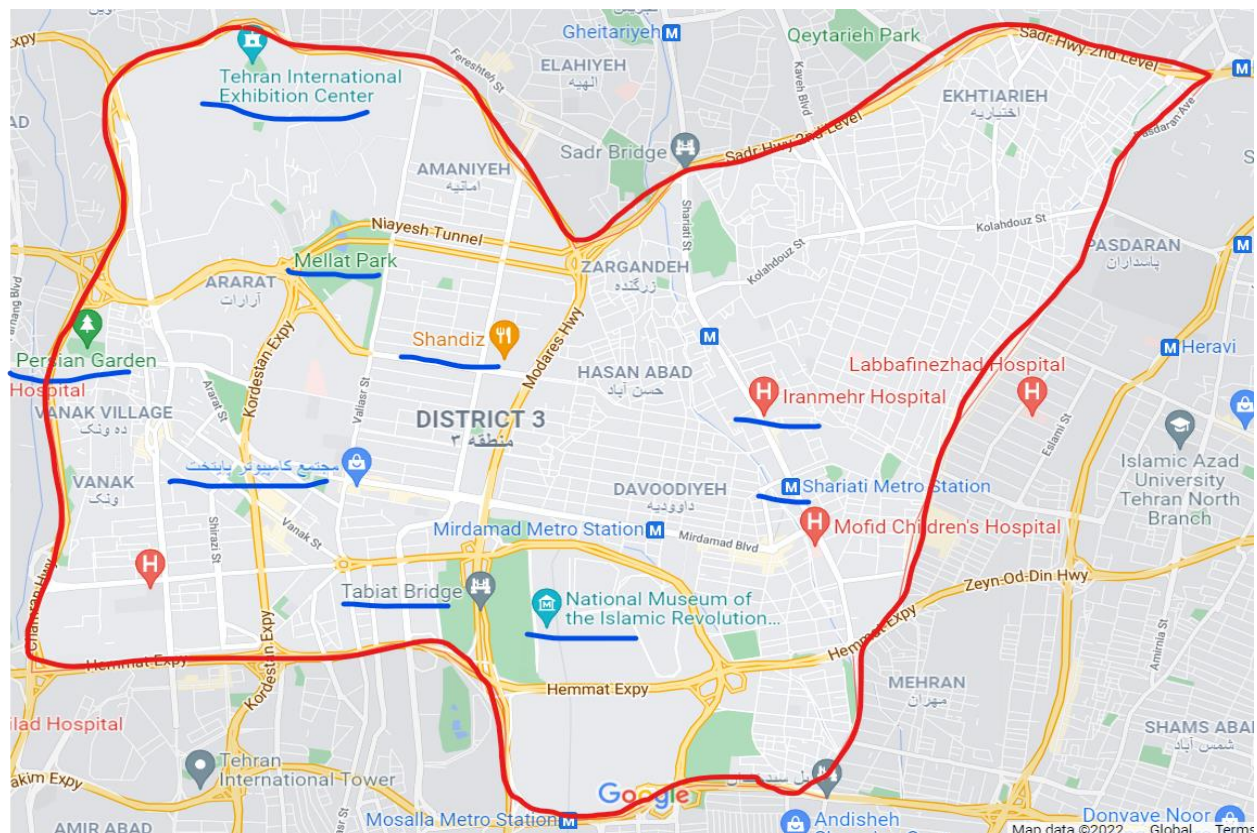
در ابتدا تنها نواحی مربوط به منطقه 3 و 6 را از روی google - map مشخص می‌کنیم .



شکل 1-1: مرز بین دو ناحیه 3 و 6

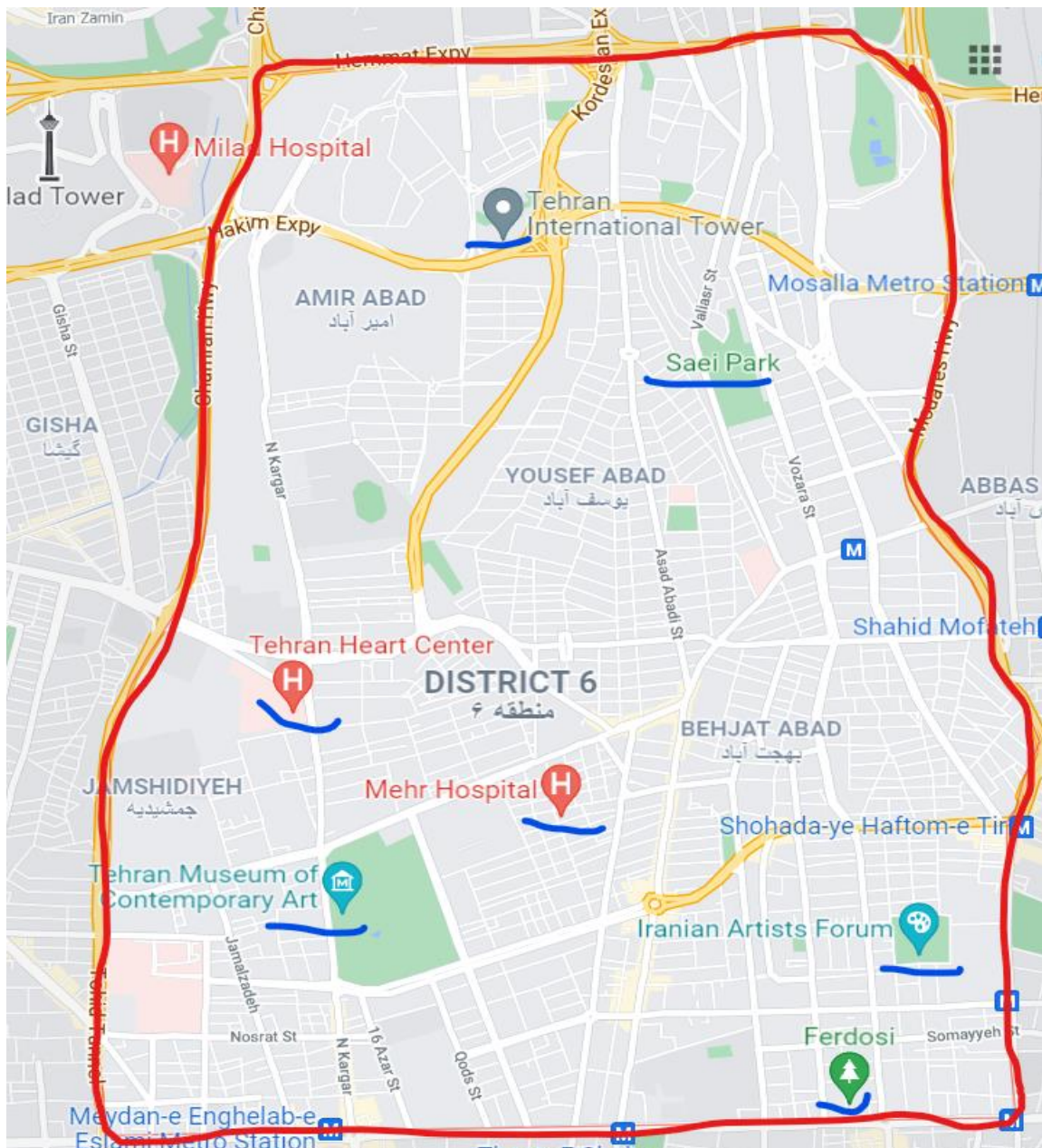
حال به ترتیب برای هر منطقه ، مکان ها (nodes) مربوطه را به طور دلخواه تشکیل می‌دهیم :

منطقه 3:



شکل 1-2: مکان های انتخابی برای منطقه 3

همانطور که مشخص است ، در مجموع 9 مکان مختلف را به عنوان 9 گره اولیه انتخاب کردیم .



شکل 1-3: مکان های انتخابی برای منطقه 6

همانطور که مشخص است ، در مجموع 7 مکان مختلف را به عنوان 7 گره ثانویه انتخاب کردیم .

پس در مجموع ، گراف ما دارای 16 گره می باشد .

حال طبق خواسته مساله ، فایل اکسلی با ستون های ویژگی داده شده ، تشکیل می دهیم :

Place_index	Place_name	coordinates	Neighbors_indice	distance(km),time(min)	Neighbors_weight
1	TIEC	35.79105095628615, 51.40298608200715	2,3	(5.3,9),(5.8,14)	14.3,19.8
2	Mellat Park	35.77823306850121, 51.410100213125865	1,3,4,5	(5.3,9),(4.6,11),(3.5,12),(3.1,7)	14.3,15.6,15.5,10.1
3	Persian Garden	35.772070289055634, 51.39034401749359	1,2,5	(5.8,14),(4.6,11),(4.7,13)	19.8,15.6,17.7
4	Shandiz	35.771328226614926, 51.42222781465077	2,5,9	(3.5,12),(2.8,9),(5.7,11)	15.5,11.8,16.7
5	Payetakht	35.76284427056937, 51.41098196750635	2,4,6	(3.1,7),(2.8,9),(4.1,7)	10.1,11.8,11.1
6	Tabiat Bridge	35.75450272407956, 51.42044824082823	7,10	(6,9),(6.1,12)	15,18.1
7	NMIR	35.75331515628775, 51.42552061152774	6,8,13	(6,9),(8.5,19),(8.4,13)	15.27.5,21.4
8	Shariati-MS	35.76274691634308, 51.44422539118962	7,9	(8.5,19),(1.6,6)	27.5,7.6
9	Iranmehr Hospital	35.76747896011174, 51.44212767701501	4,8,14	(5.7,11),(1.6,6),(11.8,17)	16.7,7.6,28.8
10	TIT	35.74253787444323, 51.39900388114728	6,11,15	(6.1,12),(6.2,12),(7.4,10)	18.1,18.2,17.4
11	Saei Park	35.73646760019867, 51.410621577994576	10,12,16	(6.2,12),(8.3,20),(4.3,9)	13.3,18.2,28.3
12	Tehran Heart Center	35.72059641498209, 51.387987961736016	13,14	(3.8,9),(2.8,10)	12.8,12.8
13	Mehr Hospital	35.715802449684205, 51.4021237744957	7,12,14,15	(8.4,13),(3.8,9),(1.7,6),(3,8)	21.4,12.8,7.7,11
14	TMCA	35.71128470164337, 51.390615739147286	9,12,13	(11.8,17),(2.8,10),(1.7,6)	28.8,12.8,7.7
15	IAF	35.709431397820964, 51.42086269890727	10,13,16	(7.4,10),(3,8),(1.3,5)	17.4,11.6,3
16	Ferdosi Park	35.702346546840815, 51.41716535126819	11,15	(4.3,9),(1.3,5)	13.3,6.3

شکل 1-4 : وارد کردن اطلاعات مساله در فایل اکسل

طبق جدول بالا ، ستون coordinates ، مختصات مکان مورد نظر را از منظر جغرافیایی تشکیل می دهد . همچنین برای هر مکان ، همسایه های متناظر با آن درون یک سلول بوسیله ویرگول از هم جدا شده اند .

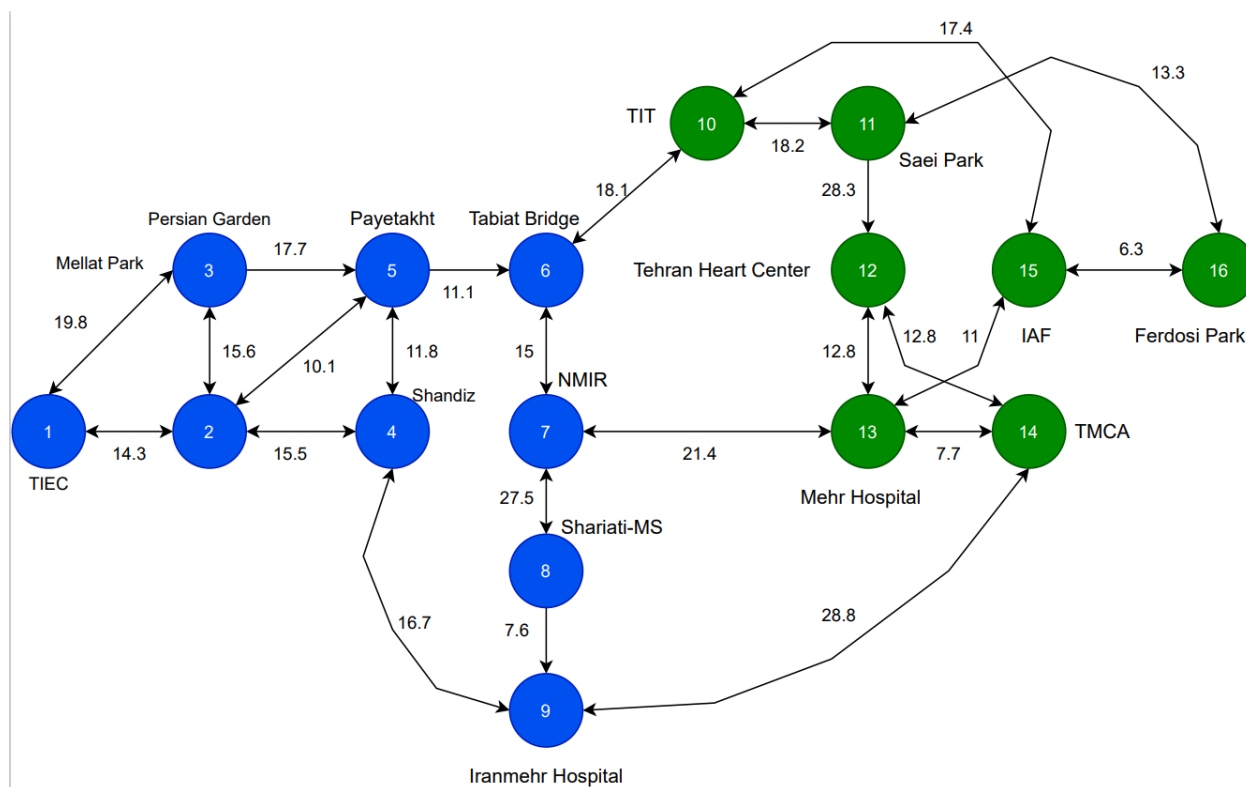
در ستون distance, time ، برای هر مکان و همسایه متناظرش ، مسافت لازم و زمان پیشنهادی توسط google-map آورده شده است.

در نهایت در ستون آخر ، وزن بین هر دو گره را توسط جمع مسافت لازم و زمان پیشنهادی بدست می آوریم .

طبیعتا معیار های متفاوتی برای تعریف وزن می تواند وجود داشته باشد ولی چون به دنبال مینیمم کردن مسافت های پیمایش شده هستیم ، وزن بین هر دو گره با زمان پیشنهادی و مسافت لازم باید رابطه مستقیم داشته باشد .

:Model Graph

حال در این مرحله باید ، اطاعات کد شده در فایل اکسل را در قالب یک گراف مدل کنیم . به کمک سایت DrawIO ، مدل نهایی گراف ما به صورت زیر خواهد بود :



شکل 1-2 : مدل نهایی گراف مساله

در گراف بالا ، دایره های با رنگ آبی مربوط به منطقه 3 و داده های با رنگ سبز مربوط به منطقه 6 هستند.

:Minimum Cost Routing

در این قسمت ، مساله را به کمک الگوریتم [Dijkstra](#) حل میکنیم .

قید های مساله این خواهد بود که از هر node به تعداد بیشتر از یک بار نباید عبور شود و هدف کمینه کردن جمع وزن های گرفته شده بین دو نقطه مبدا و مقصد می باشد .

در ابتدا فایل اکسل قسمت اول را با پایتون می خوانیم و سپس کدی می نویسیم که بتواند برای هر دو گره که همسایه هستند ، tuple زیر را برگرداند :

$(node1, node2, between_{weights})$

در نهایت با اجرای الگوریتم ، نمونه خروجی زیر را خواهیم داشت :

```
find_optimal_rout(origin,destination)
```

```
the origin is Payetakht and it's coordinate on map is 35.76284427056937, 51.41098196750635
```

```
the destination is TMCA and it's coordinate on map is 35.71128470164337, 51.390615739147286
```

```
The optimal sequence indices between cities are : ['5', '6', '7', '13', '14']
```

```
The optimal sequence between cities are : ['Payetakht', 'Tabiat Bridge', 'NMIR', 'Mehr Hospital', 'TMCA']
```

```
The minimum cost is 55.2
```

شکل 3-1 : نمونه خروجی حاصل از اجرای الگوریتم یاد شده

که با مقایسه با گراف قسمت دوم ، نتیجه همخوانی دارد :

