In [11]:

```
import pandas as pd
import matplotlib.pyplot as plt
```

In [12]:

```
file1 = r'./subway_raw.csv'
sub = pd.read_csv(file1)
sub.head()
```

Out[12]:

| | 사용일자 | 요일 | 노선명 | 역명 | 승차총승객수 | 하차총승객수 | 승하차총승객수 | 연월 | 월일 | 등록일자 |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 2019-03-01 | 목 | 4호선 | 서울역 | 62588.0 | 64794.0 | 127382.0 | 2019-03 | 03-01 | 20190304 |
| **1** | 2019-03-01 | 목 | 1호선 | 신설동 | 1335.0 | 1375.0 | 2710.0 | 2019-03 | 03-01 | 20190304 |
| **2** | 2019-03-01 | 목 | 6호선 | 보문 | 1208.0 | 1271.0 | 2479.0 | 2019-03 | 03-01 | 20190304 |
| **3** | 2019-03-01 | 목 | 4호선 | 성신여대입구 | 3231.0 | 3650.0 | 6881.0 | 2019-03 | 03-01 | 20190304 |
| **4** | 2019-03-01 | 목 | 우이신설경전철 | 정릉 | 3620.0 | 2869.0 | 6489.0 | 2019-03 | 03-01 | 20190304 |

# 1. 2019.01~06 중에 언제 지하철을 가장 많이 이용했을까?(기준: 승하차총승객수)

In [13]:

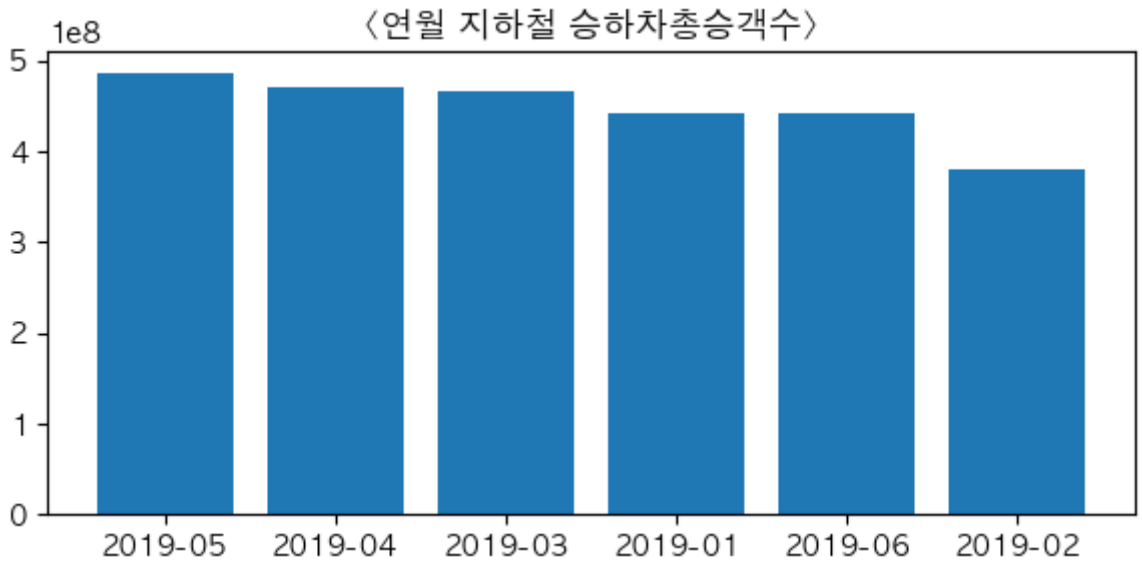
```
submonth = sub.groupby('연월')[['승하차총승객수']].sum()
submonth
```

Out[13]:

| 연월 | 승하차총승객수 |
|---|---|
| **2019-01** | 442746389.0 |
| **2019-02** | 379836010.0 |
| **2019-03** | 466692826.0 |
| **2019-04** | 470934348.0 |
| **2019-05** | 485718557.0 |
| **2019-06** | 442210635.0 |

In [48]:

```
import numpy as np
x = np.arange(len(submonth.index))
plt.rcParams['font.family'] = 'AppleGothic'
plt.figure(figsize=(7, 3))
plt.bar(x,submonth['승하차총승객수'].sort_values(ascending=False))
plt.xticks(x, submonth['승하차총승객수'].sort_values(ascending=False).index)
plt.title("<연월 지하철 승하차총승객수>")
plt.show()
```




연월로 그룹핑하여, 승하차총승객수의 합을 구했다.
그래프를 봤을때 승하차총승객수가 최대가 되는 연월은 **'2019-05'**이다.

# 2. 가설)1월~6월 중에 5월에 지하철 승객수가 많다?(기준: 승하차총승객수

In [15]:

```
submonth = sub.groupby('연월')[['승하차총승객수']].sum()
submonth[submonth['승하차총승객수']==submonth['승하차총승객수'].max()]
```

Out[15]:

| 연월 | 승하차총승객수 |
|---|---|
| **2019-05** | 485718557.0 |

'연월'로 그룹핑하여, 승하차총승객수의 합을 구했다.
이때 승하차총승객수가 최대가 되는 연월은 **'2019-05'**이다. 따라서 가설은 옳다.
~~사실 1,2번 같은거 같긴한데.. 다르게 구해봤어여..~~

# 3. 가설) 요일 중에서 목요일에 지하철 승객수가 많다?(기준: 승하차 총승객수)

In [16]:

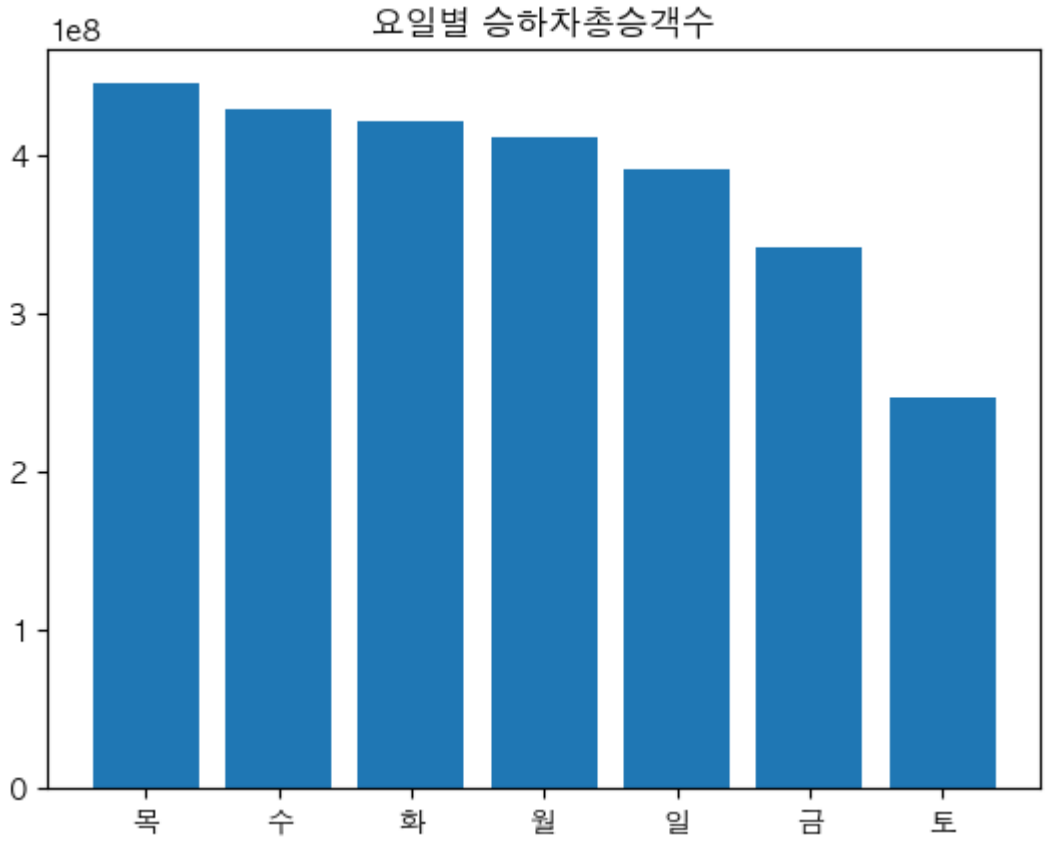
```
subday = sub.groupby('요일')[['승하차총승객수']].sum()
subday
```

Out[16]:

| 요일 | 승하차총승객수 |
|---|---|
| 금 | 341950018.0 |
| 목 | 445310717.0 |
| 수 | 428684383.0 |
| 월 | 411979965.0 |
| 일 | 391555551.0 |
| 토 | 247523995.0 |
| 화 | 421134136.0 |

In [17]:

```
x = [0, 1, 2, 3, 4, 5, 6]
plt.rcParams['font.family'] = 'AppleGothic'
plt.bar(x, subday['승하차총승객수'].sort_values(ascending=False))
plt.xticks(x, subday['승하차총승객수'].sort_values(ascending=False).index)
plt.title('요일별 승하차총승객수')
plt.show()
```



'요일'로 그룹핑하여, 승하차총승객수의 합을 구했다.
이때 승하차총승객수가 최대가 되는 요일은 **'목'**이다.
따라서 가설은 옳다.

---

# 4. 연월 각각에 대해 일자별(월일별) 승하차총승객수 그래프 그려볼까요?(pointplot)

In [18]:

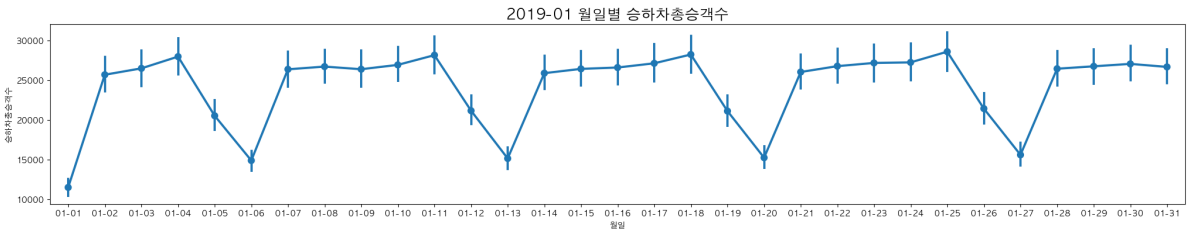
```
import seaborn as sns
```

In [19]:

```
submonth1=sub[sub['연월']=='2019-01']
plt.figure(figsize=(25, 4))
plt.rcParams['font.family'] = 'AppleGothic'
plt.title("2019-01 월일별 승하차총승객수", fontsize=18)
sns.pointplot(x='월일', y='승하차총승객수', data=submonth1)
```

Out[19]:

```
<AxesSubplot: title={'center': '2019-01 월일별 승하차총승객수'}, xlabel='월
일', ylabel='승하차총승객수'>
```

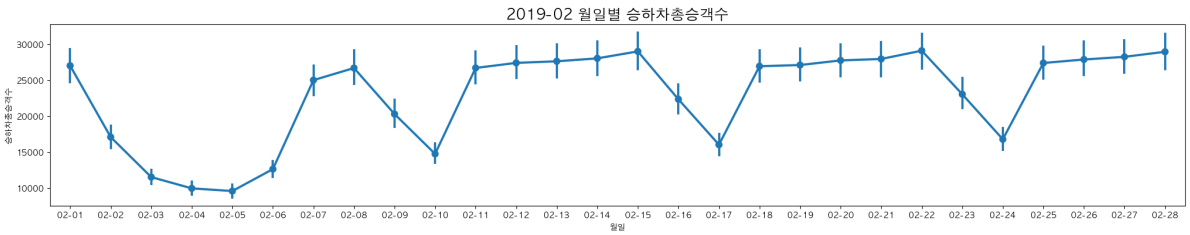



In [20]:

```
submonth2=sub[sub['연월']=='2019-02']
plt.figure(figsize=(25, 4))
plt.rcParams['font.family'] = 'AppleGothic'
plt.title("2019-02 월일별 승하차총승객수", fontsize=18)
sns.pointplot(x='월일', y='승하차총승객수', data=submonth2)
```

Out[20]:

```
<AxesSubplot: title={'center': '2019-02 월일별 승하차총승객수'}, xlabel='월
일', ylabel='승하차총승객수'>
```

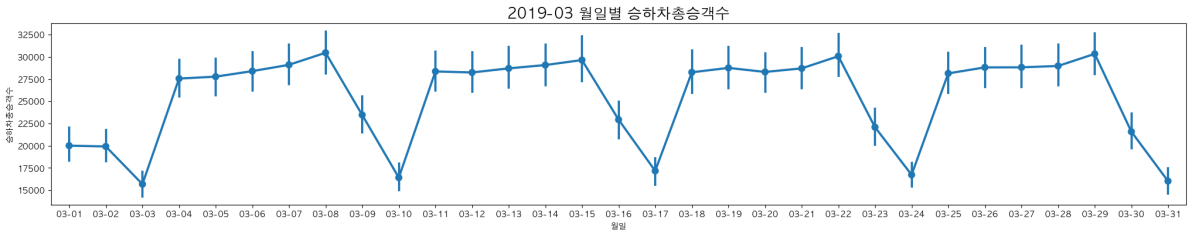


In [21]:

```
submonth3=sub[sub['연월']=='2019-03']
plt.figure(figsize=(25, 4))
plt.rcParams['font.family'] = 'AppleGothic'
plt.title("2019-03 월일별 승하차총승객수", fontsize=18)
sns.pointplot(x='월일', y='승하차총승객수', data=submonth3)
```

Out[21]:

```
<AxesSubplot: title={'center': '2019-03 월일별 승하차총승객수'}, xlabel='월
일', ylabel='승하차총승객수'>
```

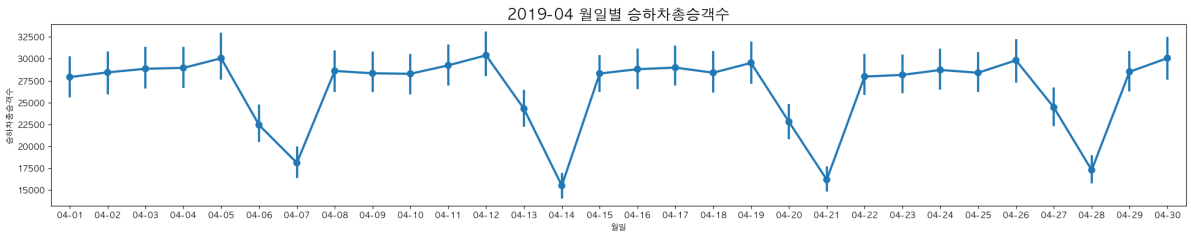



In [22]:

```
submonth4=sub[sub['연월']=='2019-04']
plt.figure(figsize=(25, 4))
plt.rcParams['font.family'] = 'AppleGothic'
plt.title("2019-04 월일별 승하차총승객수", fontsize=18)
sns.pointplot(x='월일', y='승하차총승객수', data=submonth4)
```

Out[22]:

```
<AxesSubplot: title={'center': '2019-04 월일별 승하차총승객수'}, xlabel='월
일', ylabel='승하차총승객수'>
```

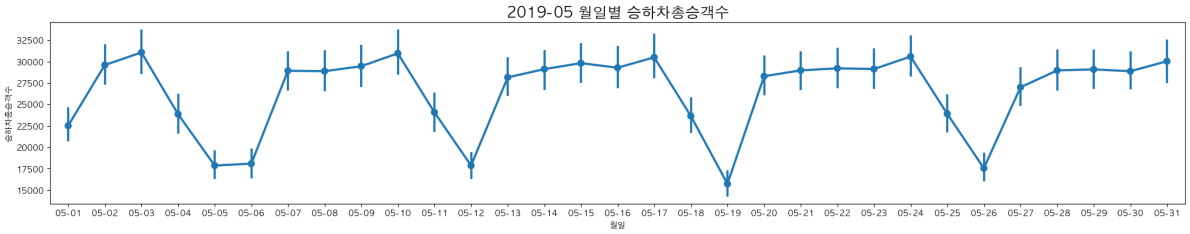


In [23]:

```
submonth5=sub[sub['연월']=='2019-05']
plt.figure(figsize=(25, 4))
plt.rcParams['font.family'] = 'AppleGothic'
plt.title("2019-05 월일별 승하차총승객수", fontsize=18)
sns.pointplot(x='월일', y='승하차총승객수', data=submonth5)
```

Out[23]:

```
<AxesSubplot: title={'center': '2019-05 월일별 승하차총승객수'}, xlabel='월
일', ylabel='승하차총승객수'>
```

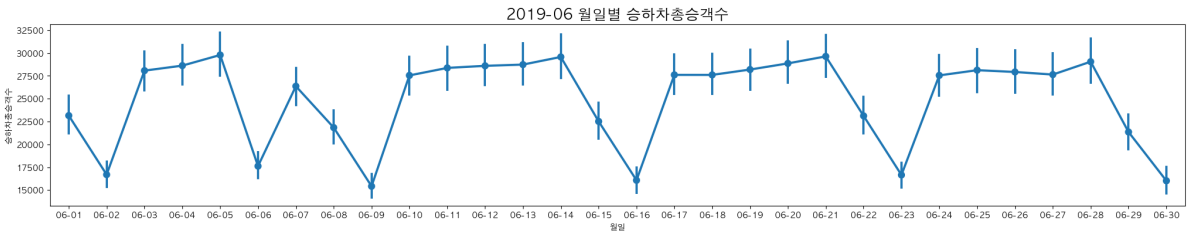



In [24]:

```
submonth6=sub[sub['연월']=='2019-06']
plt.figure(figsize=(25, 4))
plt.rcParams['font.family'] = 'AppleGothic'
plt.title("2019-06 월일별 승하차총승객수", fontsize=18)
sns.pointplot(x='월일', y='승하차총승객수', data=submonth6)
```

Out[24]:

```
<AxesSubplot: title={'center': '2019-06 월일별 승하차총승객수'}, xlabel='월
일', ylabel='승하차총승객수'>
```




# 5. 가장 승객이 많이 타는 승차역은?

In [25]:

```
substa = sub.groupby('역명')[['승차총승객수']].sum()
substa=substa.sort_values(by='승차총승객수',ascending=True)
plt.figure(figsize=(10,100))
plt.rcParams['font.family'] = 'AppleGothic'
plt.title("역별 승차총승객수",fontsize=20)
substa.unstack(0).plot.barh()
```

Out[25]:

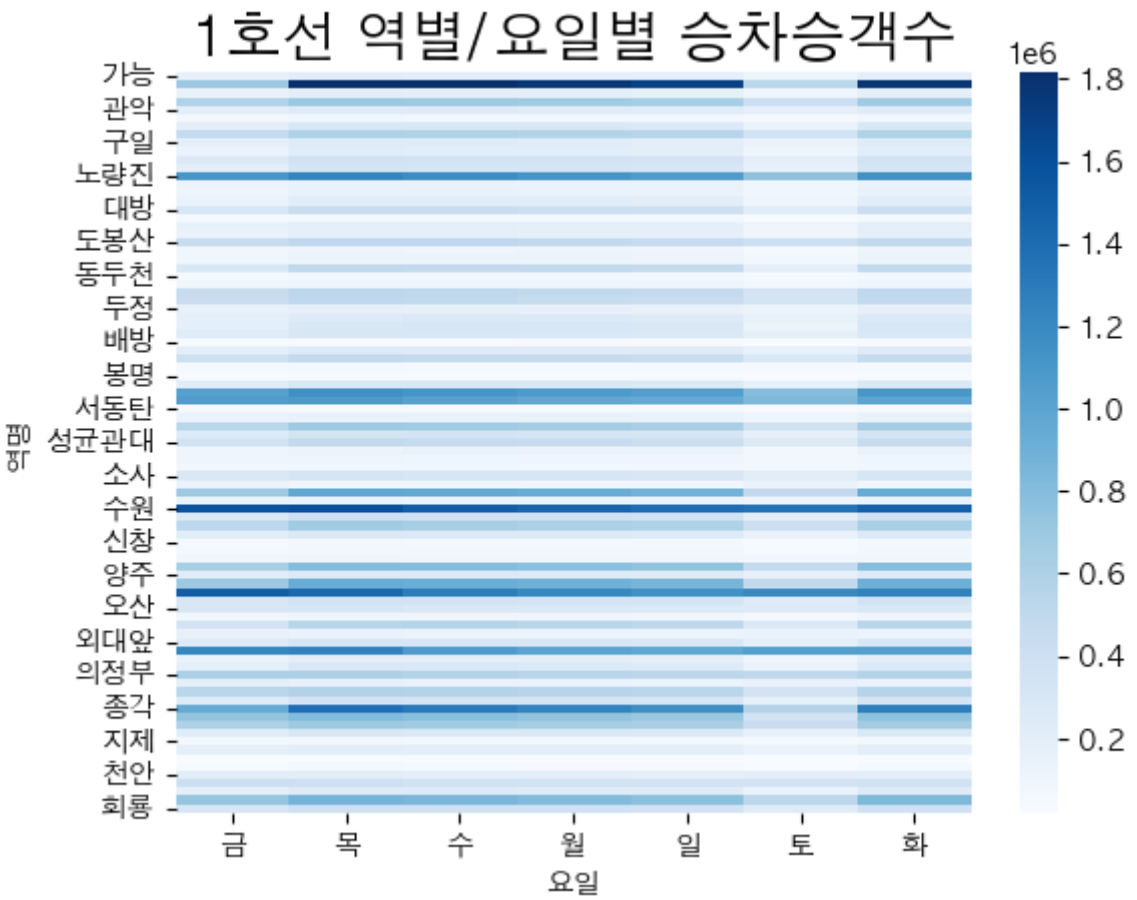
```
<AxesSubplot: title={'center': '역별 승차총승객수'}, ylabel='None,역명'>
```

'역명'으로 그룹핑하여, 승차총승객수의 합을 구했다. 그리고 승차총승객수를 기준으로 내림차순 정렬을 해줬다
이때 승차총승객수가 최대가 되는 역은 **'잠실'**이다.

# 6. 노선별로 역별/요일별 승차승객수를 비교해볼 수 있을까?(1~9호선, 역별/요일별 heatmap)

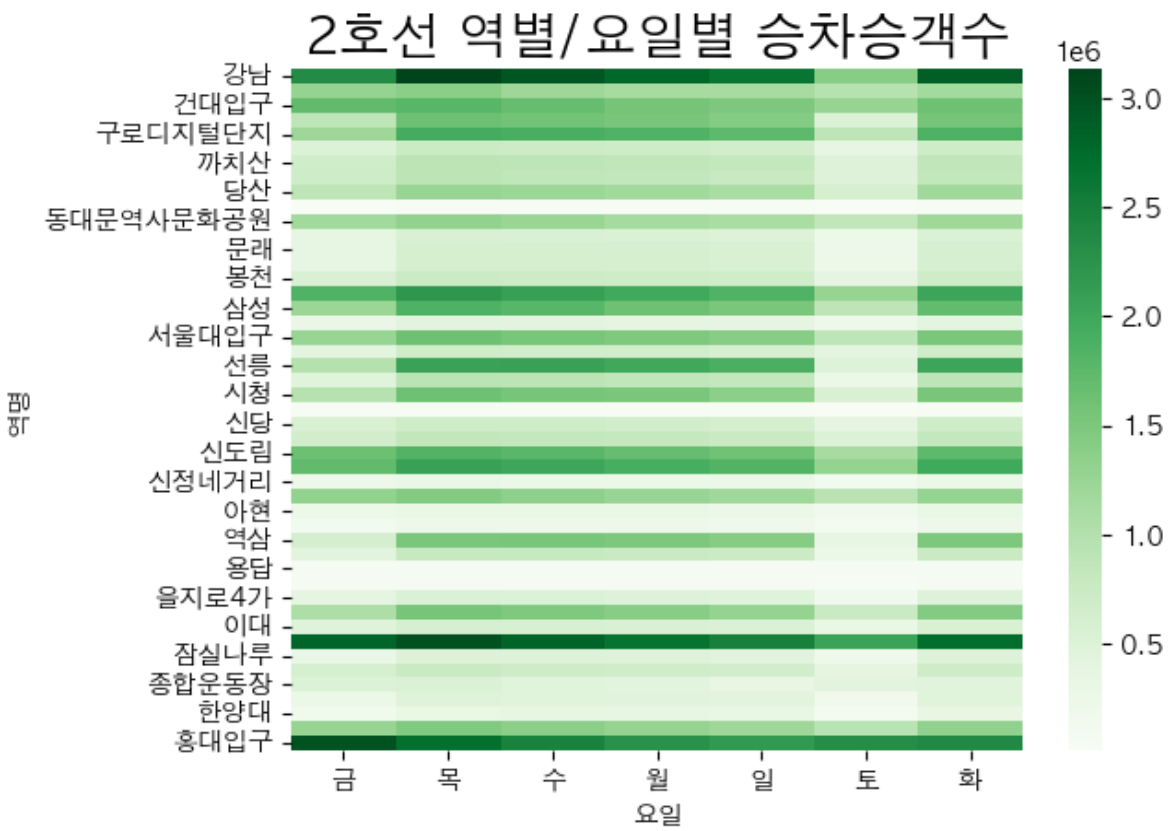
In [26]:

```
sub1=sub[sub['노선명']=='1호선']
df1 = sub1.groupby(['역명', '요일'])[['승차총승객수']].sum()
df1 = pd.pivot_table(data=df1, index='역명', columns='요일', values='승차총승객수', aggfu
sns.heatmap(df1, cmap='Blues')
plt.title('1호선 역별/요일별 승차승객수', fontsize=20)
plt.show()
```



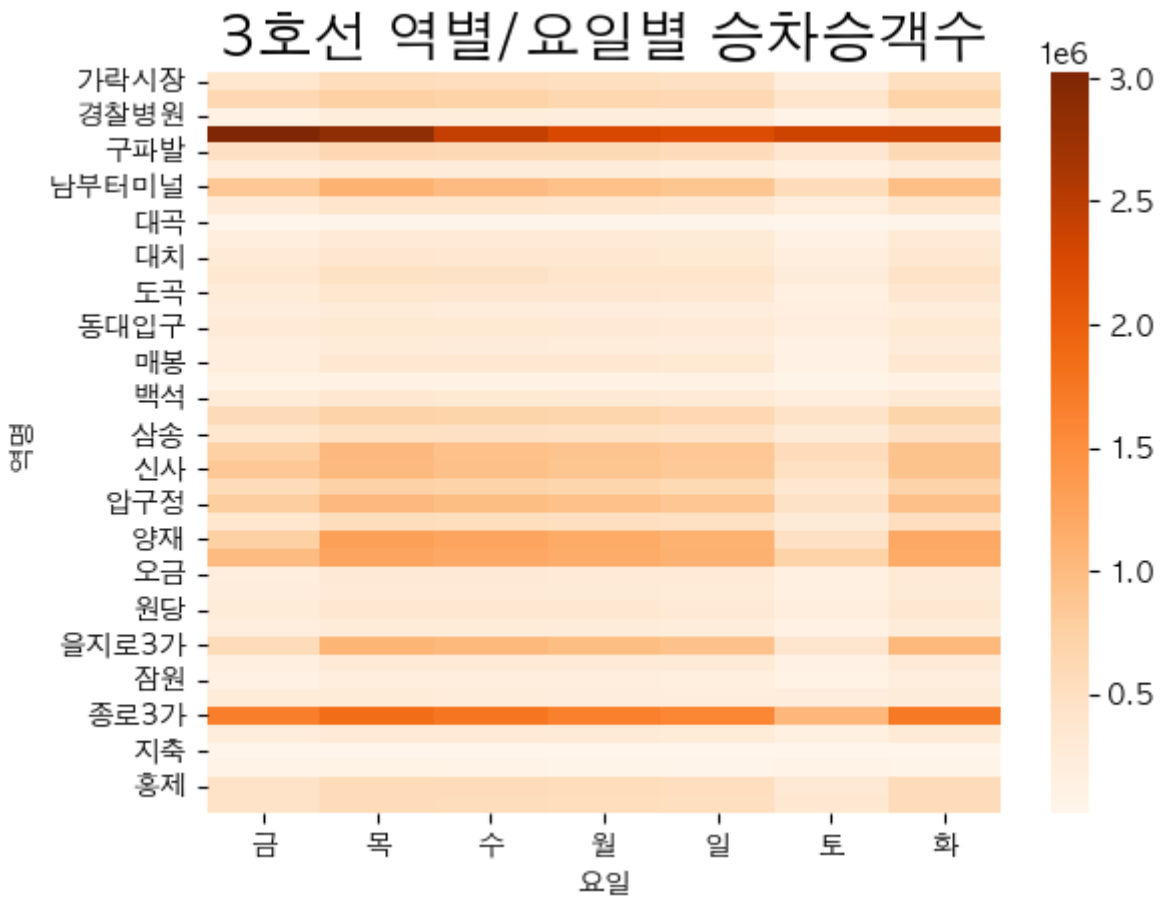
In [27]:

```
sub2 = sub[sub['노선명']=='2호선']
df2 = sub2.groupby(['역명', '요일'])[['승차총승객수']].sum()
df2 = pd.pivot_table(data=df2, index='역명', columns='요일', values='승차총승객수', aggfu
ax=sns.heatmap(df2, cmap='Greens')
plt.title('2호선 역별/요일별 승차승객수', fontsize=20)
plt.show()
```



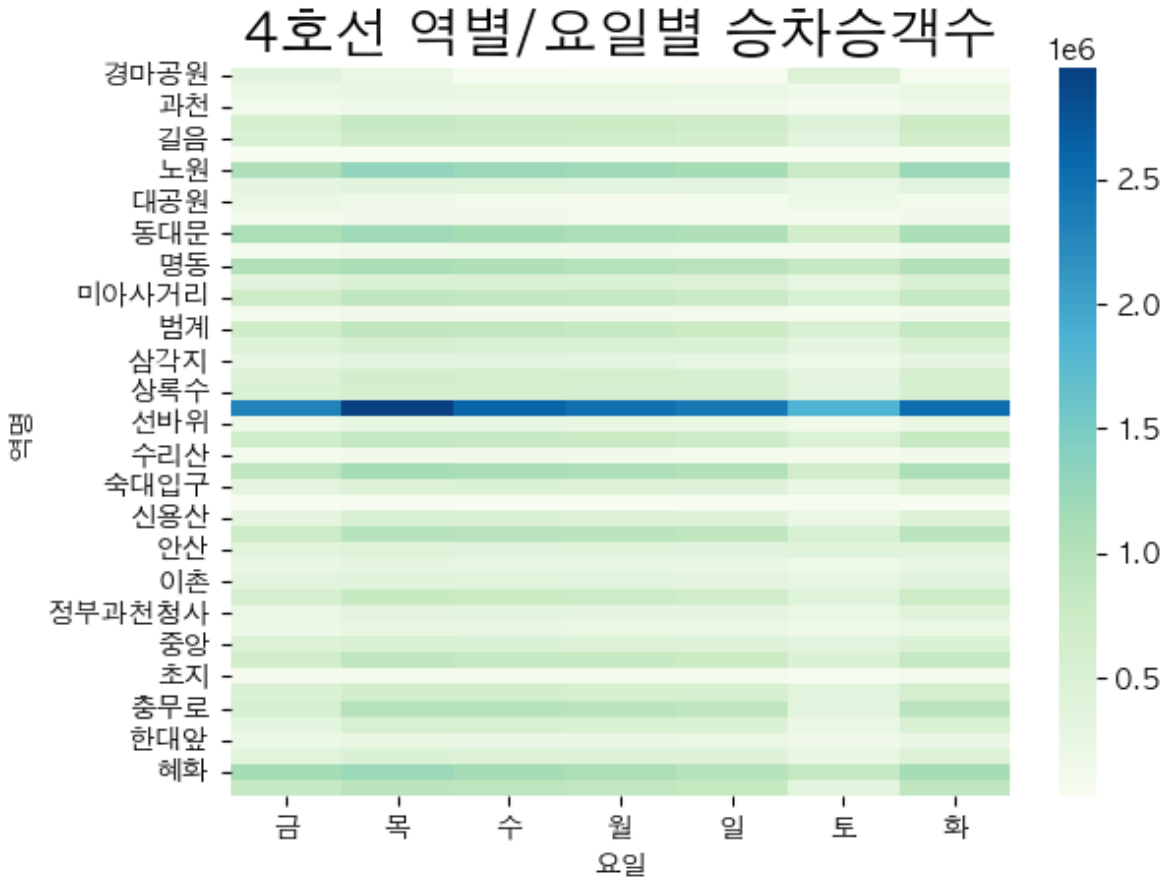
In [28]:

```
sub3 = sub[sub['노선명']=='3호선']
df3 = sub3.groupby(['역명', '요일'])[['승차총승객수']].sum()
df3 = pd.pivot_table(data=df3, index='역명', columns='요일', values='승차총승객수', aggfu
ax=sns.heatmap(df3, cmap='Oranges')
plt.title('3호선 역별/요일별 승차승객수', fontsize=20)
plt.show()
```



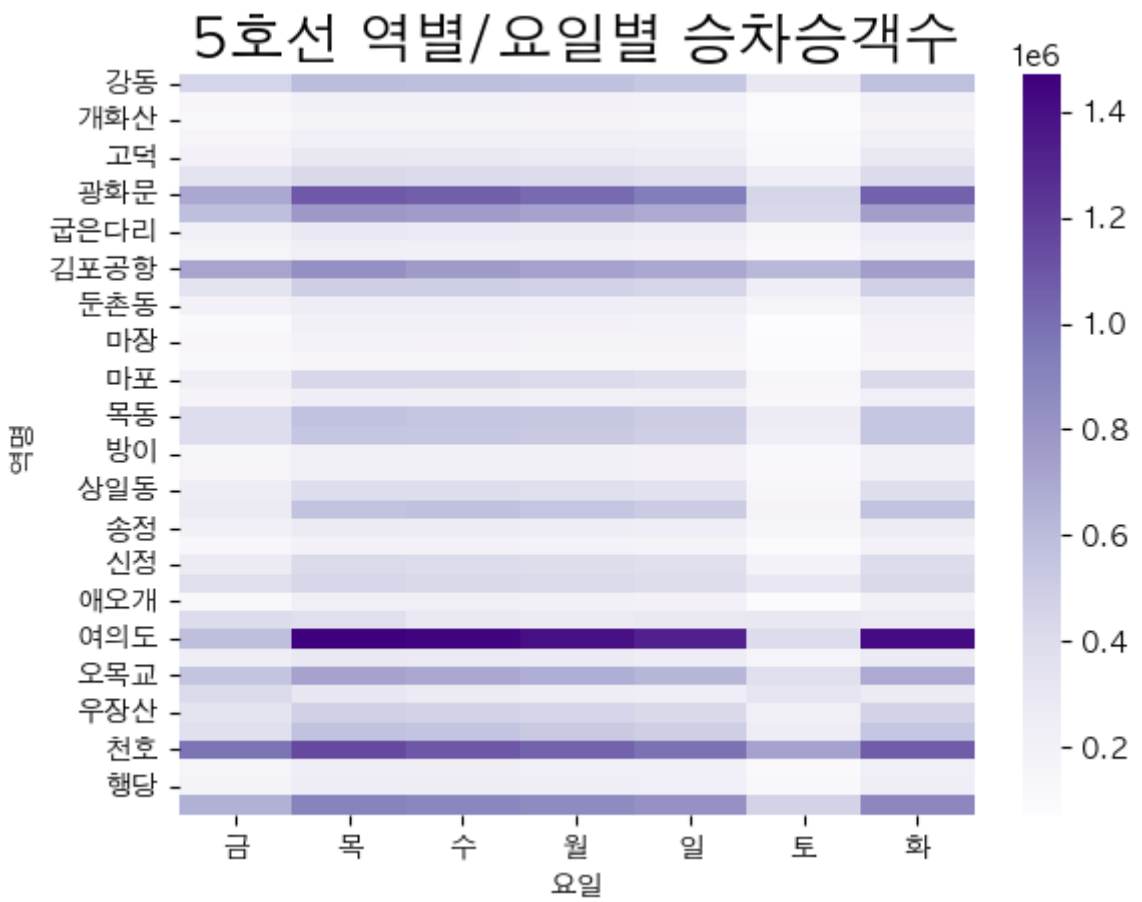
In [29]:

```
sub4 = sub[sub['노선명']=='4호선']
df4 = sub4.groupby(['역명', '요일'])[['승차총승객수']].sum()
df4 = pd.pivot_table(data=df4, index='역명', columns='요일', values='승차총승객수', aggfu
ax=sns.heatmap(df4, cmap='GnBu')
plt.title('4호선 역별/요일별 승차승객수', fontsize=20)
plt.show()
```



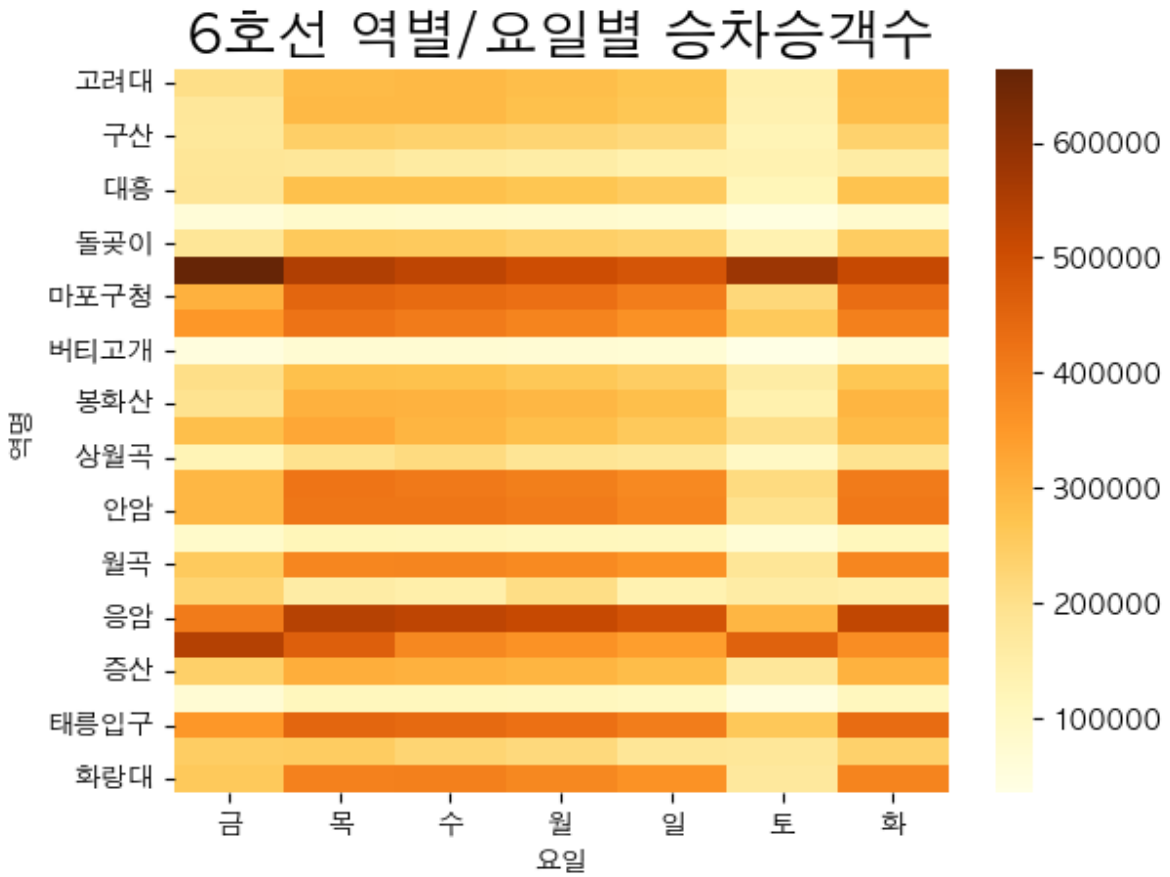
In [30]:

```
sub5 = sub[sub['노선명']=='5호선']
df5 = sub5.groupby(['역명', '요일'])[['승차총승객수']].sum()
df5 = pd.pivot_table(data=df5, index='역명', columns='요일', values='승차총승객수', aggfu
ax=sns.heatmap(df5, cmap='Purples')
plt.title('5호선 역별/요일별 승차승객수', fontsize=20)
plt.show()
```



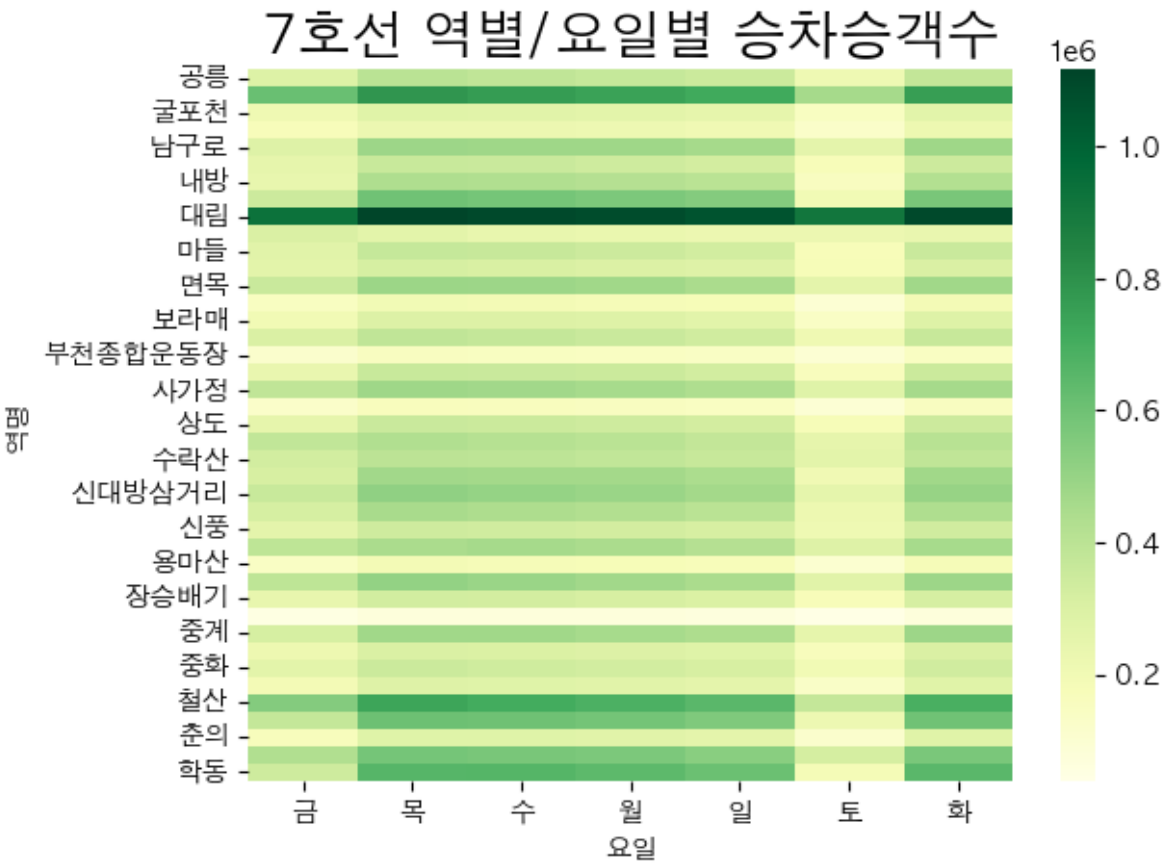
In [31]:

```
sub6 = sub[sub['노선명']=='6호선']
df6 = sub6.groupby(['역명', '요일'])[['승차총승객수']].sum()
df6 = pd.pivot_table(data=df6, index='역명', columns='요일', values='승차총승객수', aggfu
ax=sns.heatmap(df6, cmap='YlOrBr')
plt.title('6호선 역별/요일별 승차승객수', fontsize=20)
plt.show()
```



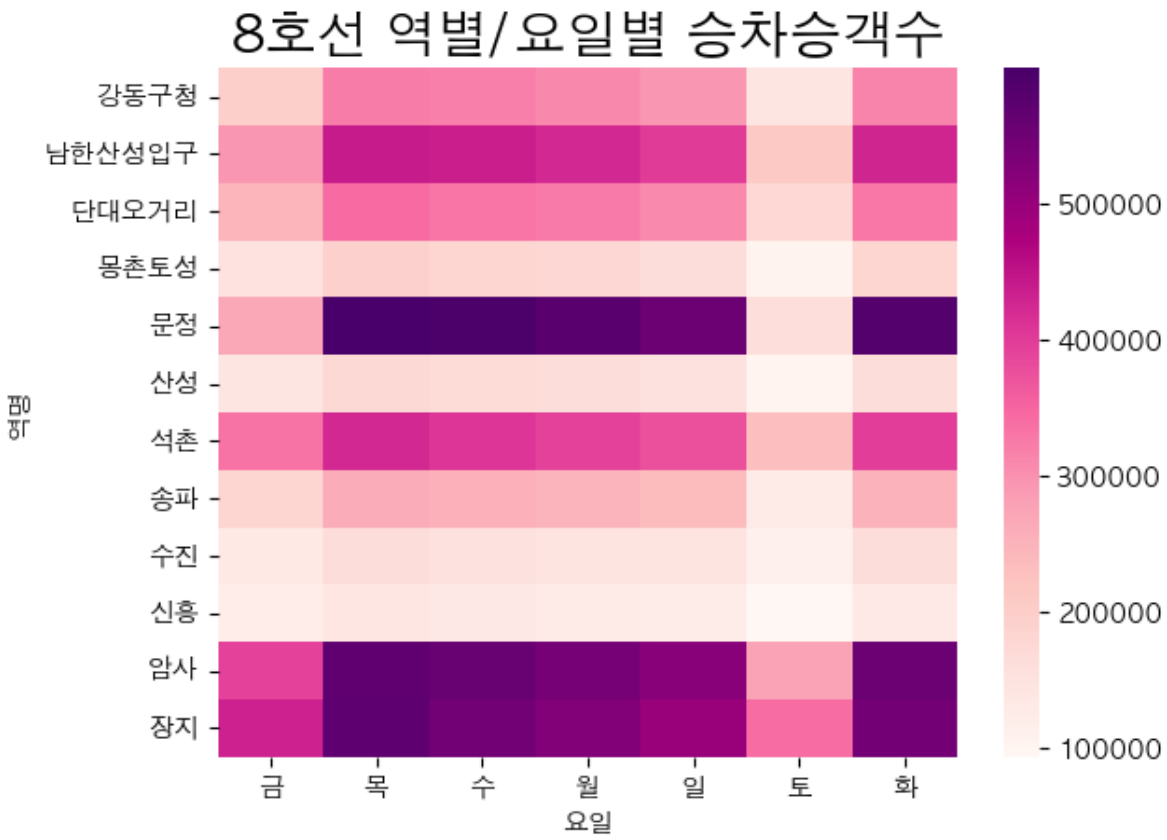
In [32]:

```
sub7 = sub[sub['노선명']=='7호선']
df7 = sub7.groupby(['역명', '요일'])[['승차총승객수']].sum()
df7 = pd.pivot_table(data=df7, index='역명', columns='요일', values='승차총승객수', aggfu
ax=sns.heatmap(df7, cmap='YlGn')
plt.title('7호선 역별/요일별 승차승객수', fontsize=20)
plt.show()
```



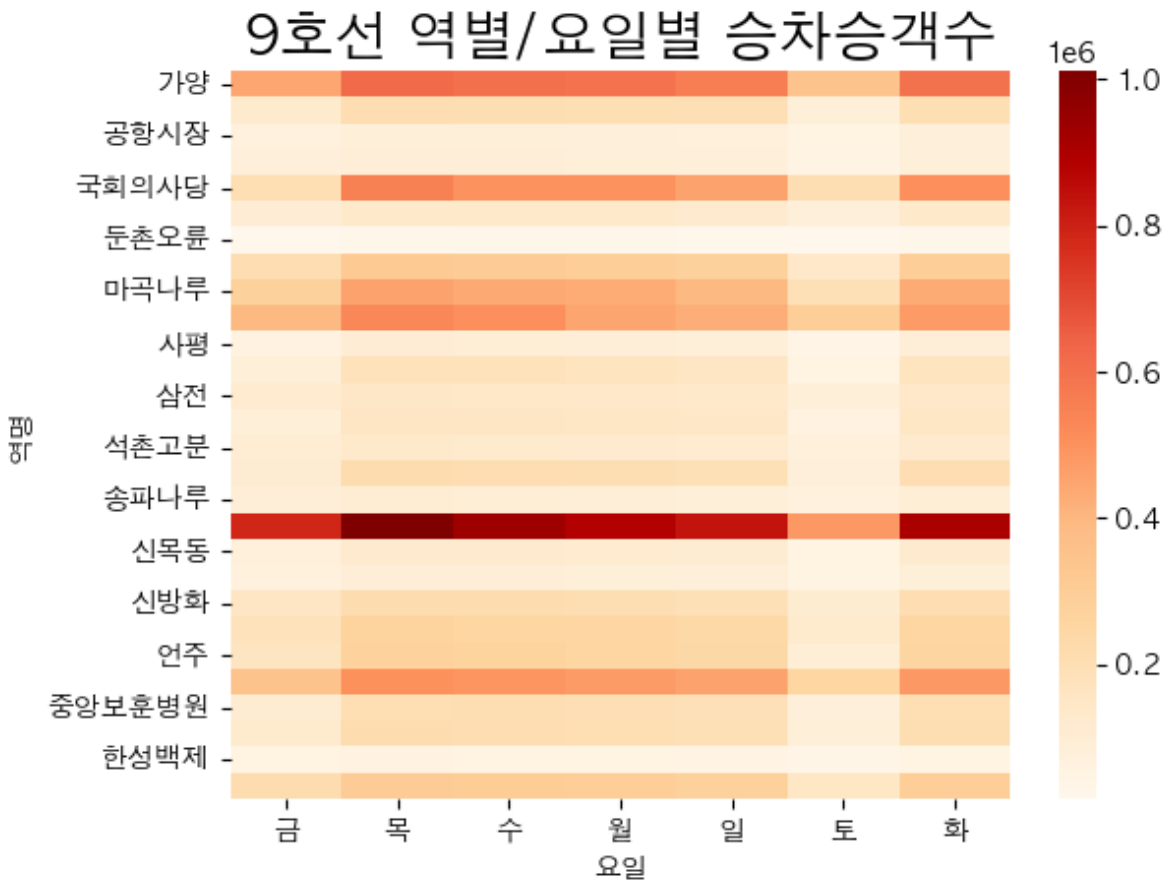
In [33]:

```
sub8 = sub[sub['노선명']=='8호선']
df8 = sub8.groupby(['역명', '요일'])[['승차총승객수']].sum()
df8 = pd.pivot_table(data=df8, index='역명', columns='요일', values='승차총승객수', aggfu
ax=sns.heatmap(df8, cmap='RdPu')
plt.title('8호선 역별/요일별 승차승객수', fontsize=20)
plt.show()
```



In [34]:

```
sub9 = sub[sub['노선명']=='9호선']
df9 = sub9.groupby(['역명', '요일'])[['승차총승객수']].sum()
df9 = pd.pivot_table(data=df9, index='역명', columns='요일', values='승차총승객수', aggfu
ax=sns.heatmap(df9, cmap='OrRd')
plt.title('9호선 역별/요일별 승차승객수', fontsize=20)
plt.show()
```



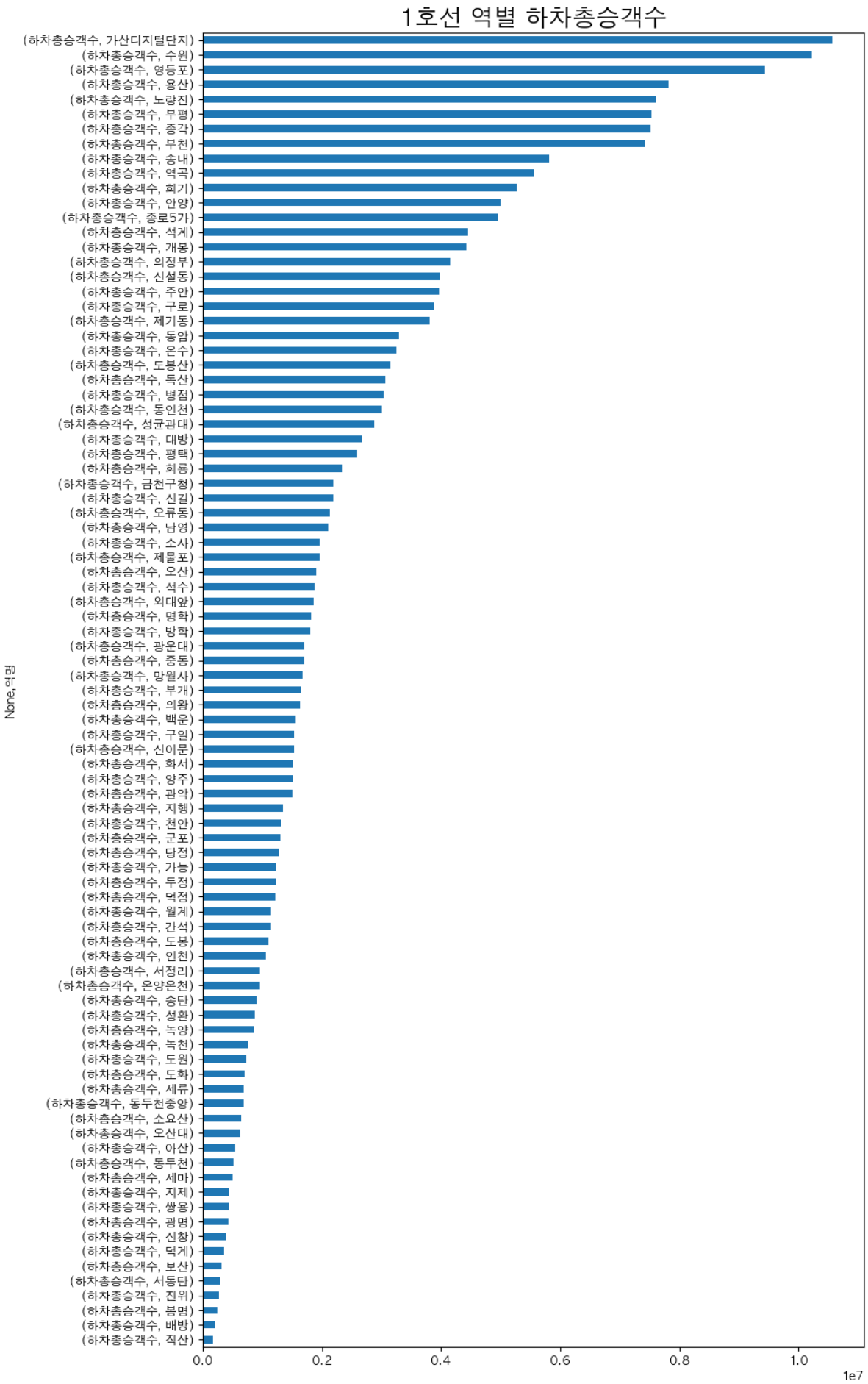

# 7. 1호선에서 가장 하차를 많이 하는 역은?(groupby)

In [50]:

```
sub1 = sub[sub['노선명']=='1호선']
sub1_hacha = sub1.groupby('역명')[['하차총승객수']].sum()
sub1_hacha = sub1_hacha.sort_values(by='하차총승객수',ascending=True)
plt.figure(figsize=(10,20))
plt.rcParams['font.family'] = 'AppleGothic'
plt.title("1호선 역별 하차총승객수",fontsize=20)
sub1_hacha.unstack(0).plot.barh()
```

Out[50]:

```
<AxesSubplot: title={'center': '1호선 역별 하차총승객수'}, ylabel='None,역
명'>
```


1호선 역별 하차총승객수
(하차총승객수, 가산디지털단지)
(하차총승객수, 수원)
(하차총승객수, 영등포)
(하차총승객수, 용산)
(하차총승객수, 노량진)
(하차총승객수, 부평)
(하차총승객수, 종각)
(하차총승객수, 부천)
(하차총승객수, 송내)
(하차총승객수, 역곡)
(하차총승객수, 회기)
(하차총승객수, 안양)
(하차총승객수, 종로5가)
(하차총승객수, 석계)
(하차총승객수, 개봉)
(하차총승객수, 의정부)
(하차총승객수, 신설동)
(하차총승객수, 주안)
(하차총승객수, 구로)
(하차총승객수, 제기동)
(하차총승객수, 동암)
(하차총승객수, 온수)
(하차총승객수, 도봉산)
(하차총승객수, 독산)
(하차총승객수, 병점)
(하차총승객수, 동인천)
(하차총승객수, 성균관대)
(하차총승객수, 대방)
(하차총승객수, 평택)
(하차총승객수, 회룡)
(하차총승객수, 금천구청)
(하차총승객수, 신길)
(하차총승객수, 오류동)
(하차총승객수, 남영)
(하차총승객수, 소사)
(하차총승객수, 제물포)
(하차총승객수, 오산)
(하차총승객수, 석수)
(하차총승객수, 외대앞)
(하차총승객수, 명학)
(하차총승객수, 방학)
(하차총승객수, 광운대)
(하차총승객수, 중동)
(하차총승객수, 망월사)
(하차총승객수, 부개)
(하차총승객수, 의왕)
(하차총승객수, 백운)
(하차총승객수, 구일)
(하차총승객수, 신이문)
(하차총승객수, 화서)
(하차총승객수, 양주)
(하차총승객수, 관악)
(하차총승객수, 지행)
(하차총승객수, 천안)
(하차총승객수, 군포)
(하차총승객수, 당정)
(하차총승객수, 가능)
(하차총승객수, 두정)
(하차총승객수, 덕정)
(하차총승객수, 월계)
(하차총승객수, 간석)
(하차총승객수, 도봉)
(하차총승객수, 인천)
(하차총승객수, 서정리)
(하차총승객수, 온양온천)
(하차총승객수, 송탄)
(하차총승객수, 성환)
(하차총승객수, 녹양)
(하차총승객수, 녹천)
(하차총승객수, 도원)
(하차총승객수, 도화)
(하차총승객수, 세류)
(하차총승객수, 동두천중앙)
(하차총승객수, 소요산)
(하차총승객수, 오산대)
(하차총승객수, 아산)
(하차총승객수, 동두천)
(하차총승객수, 세마)
(하차총승객수, 지제)
(하차총승객수, 쌍용)
(하차총승객수, 광명)
(하차총승객수, 신창)
(하차총승객수, 덕계)
(하차총승객수, 보산)
(하차총승객수, 서동탄)
(하차총승객수, 진위)
(하차총승객수, 봉명)
(하차총승객수, 배방)
(하차총승객수, 직산)
None, 역명
0.0
0.2
0.4
0.6
0.8
1.0
1e7

'노선명'이 '1호선'인 행을 추출하여 sub1에 넣어줬다.
sub1에서 '역명'으로 그룹핑하여, 하차총승객수의 합을 구했다. 그리고 내림차순으로 정렬해서 그래프로 나타내었다.
이때 하차총승객수가 최대가 되는 역은 **'가산디지털단지'**이다.

---

# 8. 2호선에서 어느 역에서 승차가 가장 많이 발생할까?(Folium역 표시)

In [36]:

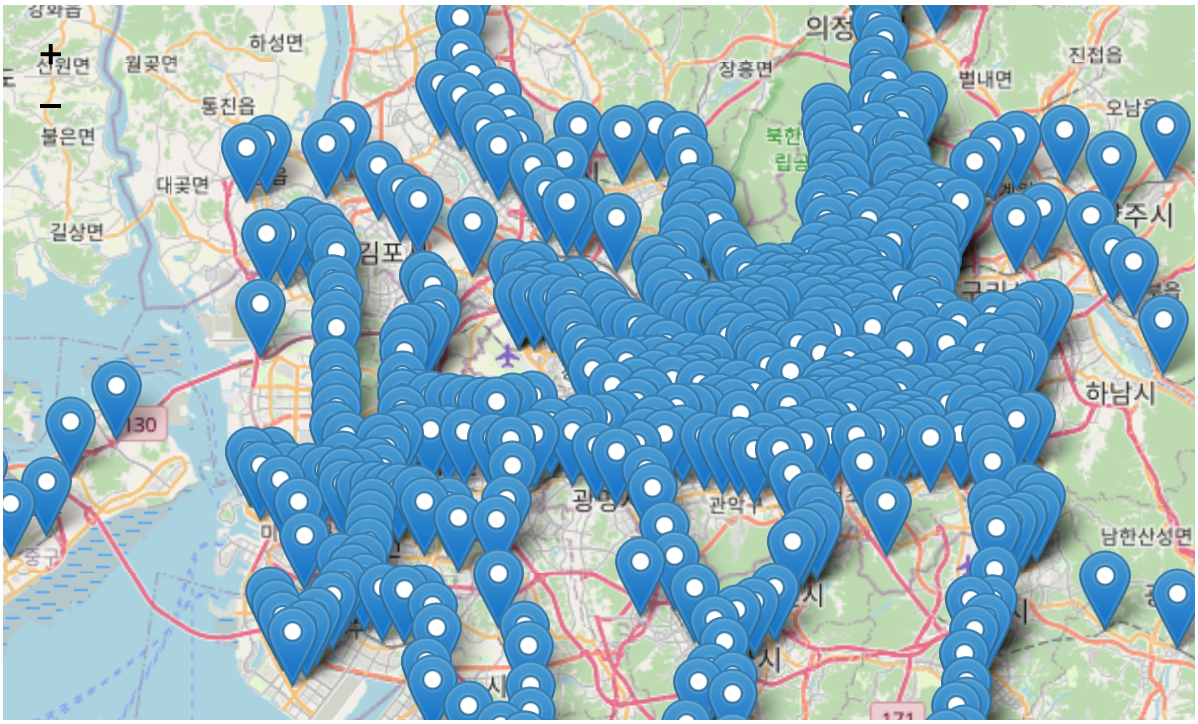
```
import folium
from folium import plugins
from folium.plugins import HeatMap
```

In [37]:

```
file2 = r'./지하철노선위경도정보2.csv'
locations = pd.read_csv(file2)

# 모든 지하철역 지도표시
m = folium.Map(location=[locations.head(1)['위도'], locations.head(1)['경도']],
               zoom_start=10,
               width=750,
               height=500
              )
for i in range(len(locations)):
    latitude = locations.at[i,'위도']
    longitude = locations.at[i,'경도']
    name = locations.at[i,'역이름']
    folium.Marker([latitude, longitude],
                  popup=name,
                  tooltip=name).add_to(m)
m
```

Out[37]:



In [38]:

```
locations2 = locations[locations['호선']=='2호선']
locations2 = locations2.reset_index(drop=False)
locations2
```

Out[38]:

| | index | 역이름 | 역지역 | 위도 | 경도 | 호선 |
|---|---|---|---|---|---|---|
| **0** | 0 | 낙성대 | 수도권 | 37.477090 | 126.963506 | 2호선 |
| **1** | 2 | 서울대입구 | 수도권 | 37.481285 | 126.952695 | 2호선 |
| **2** | 7 | 강변 | 수도권 | 37.535118 | 127.094723 | 2호선 |
| **3** | 8 | 영등포구청 | 수도권 | 37.525831 | 126.896668 | 2호선 |
| **4** | 10 | 잠실새내 | 수도권 | 37.511608 | 127.086301 | 2호선 |
| **...** | ... | ... | ... | ... | ... | ... |
| **118** | 898 | 용산역 대구2호 | 대구 | 35.849060 | 128.528804 | 2호선 |
| **119** | 899 | 이곡 | 대구 | 35.850595 | 128.515794 | 2호선 |
| **120** | 900 | 임당 | 대구 | 35.834118 | 128.740837 | 2호선 |
| **121** | 901 | 정평 | 대구 | 35.834041 | 128.728662 | 2호선 |
| **122** | 902 | 죽전 | 대구 | 35.850717 | 128.538798 | 2호선 |

123 rows × 6 columns

In [39]:

```
#2호선 지하철 표시
locations2 = locations[locations['호선']=='2호선']
locations2 = locations2.reset_index(drop=False)

m2 = folium.Map(location=[locations2.head(1)['위도'], locations2.head(1)['경도']],
                zoom_start=11,
                width=750,
                height=500
                )
for i in range(len(locations2)):
    latitude = locations2.at[i,'위도']
    longitude = locations2.at[i,'경도']
    name = locations2.at[i,'역이름']
    folium.Marker([latitude, longitude],
                  popup=name,
                  tooltip=name).add_to(m2)
```

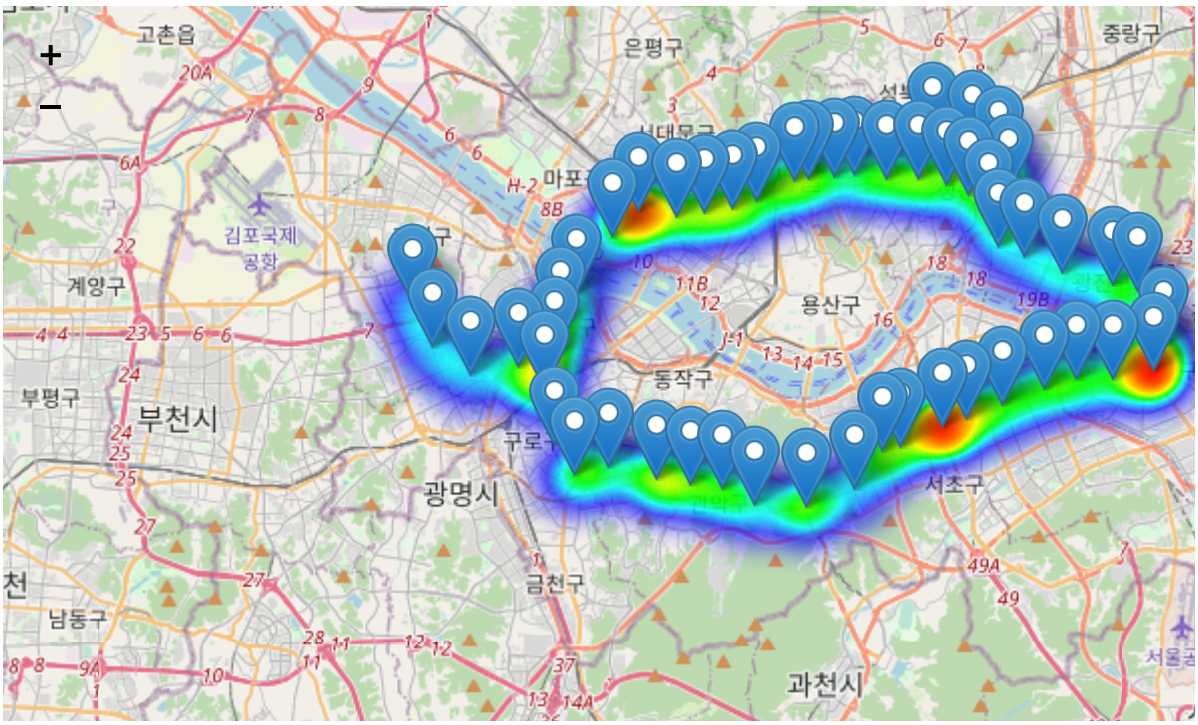
In [41]:

```
sub2 = sub[sub['노선명']=='2호선']
sub2=sub2.rename(columns={'역명':'역이름'})
sub2_seungcha = sub2.groupby('역이름')[['승차총승객수']].sum()

sub2_seungcha = pd.merge(sub2_seungcha, locations2, on='역이름')

m2.add_child(plugins.HeatMap(zip(sub2_seungcha['위도'],
                                 sub2_seungcha['경도'],
                                 sub2_seungcha['승차총승객수']), radius=18))
m2
```

Out[41]:




In [ ]: