

In [1]:

```
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
import os

%matplotlib inline
plt.rcParams['figure.figsize'] = [30, 10]

country_00 = 'US'
country_01 = 'Italy'
country_02 = 'Czechia'
country_03 = 'Germany'

country_04 = 'Austria'
country_05 = 'France'
country_06 = 'Spain'
country_07 = 'United Kingdom'
country_08 = 'Sweden'

country_09 = 'Netherlands'
country_10 = 'Switzerland'
country_11 = 'Iran'
country_12 = 'Korea, South'
country_13 = 'Singapore'
country_14 = 'Russia'
country_15 = 'Ukraine'
country_16 = 'Slovakia'
country_17 = 'China'

confirmed_cases = pd.read_csv('https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_gis/covid_19_data/covid_19_timeseries.csv')
deaths = pd.read_csv('https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_gis/covid_19_data/covid_19_timeseries.csv')
recoveries = pd.read_csv('https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_gis/covid_19_data/covid_19_timeseries.csv')

confirmed_cases.drop(columns = ['Province/State', 'Lat', 'Long'], inplace = True)
deaths.drop(columns = ['Province/State', 'Lat', 'Long'], inplace = True)
recoveries.drop(columns = ['Province/State', 'Lat', 'Long'], inplace = True)

def getSumOfDf(country, confirmed, deaths, recoveries, min_date = False):
    cs = confirmed[confirmed['Country/Region'] == country].sum()
    cs.drop('Country/Region', inplace = True)

    ds = deaths[deaths['Country/Region'] == country].sum()
    ds.drop('Country/Region', inplace = True)

    rs = recoveries[recoveries['Country/Region'] == country].sum()
    rs.drop('Country/Region', inplace = True)

    vals = list(cs.iteritems())
    earliest_date = ''

    for i, v in enumerate(vals):
        if vals[i + 1][1] > 0:
            earliest_date = v[0]
            break

    if min_date:
        earliest_date = min_date
```

```

for date, _ in cs.iteritems():
    if date == earliest_date:
        break
    else:
        cs.drop(date, inplace = True)

for date, _ in ds.iteritems():
    if date == earliest_date:
        break
    else:
        ds.drop(date, inplace = True)

for date, _ in rs.iteritems():
    if date == earliest_date:
        break
    else:
        rs.drop(date, inplace = True)

return cs, ds, rs

```

```

country_00_cases, country_00_deaths, country_00_recoveries = getSumOfDf(country_00,
country_01_cases, country_01_deaths, country_01_recoveries = getSumOfDf(country_01,
country_02_cases, country_02_deaths, country_02_recoveries = getSumOfDf(country_02,
country_03_cases, country_03_deaths, country_03_recoveries = getSumOfDf(country_03,
country_04_cases, country_04_deaths, country_04_recoveries = getSumOfDf(country_04,
country_05_cases, country_05_deaths, country_05_recoveries = getSumOfDf(country_05,
country_06_cases, country_06_deaths, country_06_recoveries = getSumOfDf(country_06,
country_07_cases, country_07_deaths, country_07_recoveries = getSumOfDf(country_07,
country_08_cases, country_08_deaths, country_08_recoveries = getSumOfDf(country_08,
country_09_cases, country_09_deaths, country_09_recoveries = getSumOfDf(country_09,
country_10_cases, country_10_deaths, country_10_recoveries = getSumOfDf(country_10,
country_11_cases, country_11_deaths, country_11_recoveries = getSumOfDf(country_11,
country_12_cases, country_12_deaths, country_12_recoveries = getSumOfDf(country_12,
country_13_cases, country_13_deaths, country_13_recoveries = getSumOfDf(country_13,
country_14_cases, country_14_deaths, country_14_recoveries = getSumOfDf(country_14,
country_15_cases, country_15_deaths, country_15_recoveries = getSumOfDf(country_15,
country_16_cases, country_16_deaths, country_16_recoveries = getSumOfDf(country_16,
country_17_cases, country_17_deaths, country_17_recoveries = getSumOfDf(country_17,

```

```

world_cases = confirmed_cases.sum().drop('Country/Region')
world_deaths = deaths.sum().drop('Country/Region')
world_recoveries = recoveries.sum().drop('Country/Region')

```

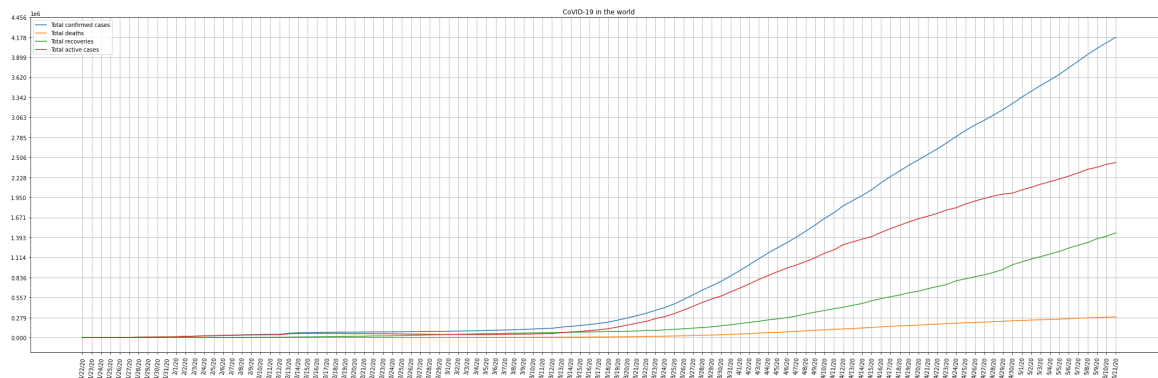
```

def plotForCountry(country, cases, deaths, recoveries, scale = 100):
    plt.plot(cases, label = 'Total confirmed cases')
    plt.plot(deaths, label = 'Total deaths')
    plt.plot(recoveries, label = 'Total recoveries')
    plt.plot(cases - recoveries - deaths, label = 'Total active cases')
    plt.title(f'CoVID-19 in {country}')
    plt.legend()
    ticks = plt.xticks(rotation = 85)
    step = round(cases.max() / (15 * scale)) * scale
    ticks = plt.yticks(np.arange(0, cases.max() + step + scale / 2, step))
    plt.rcParams.update({'font.size': 22})
    plt.tight_layout()
    plt.grid()

```

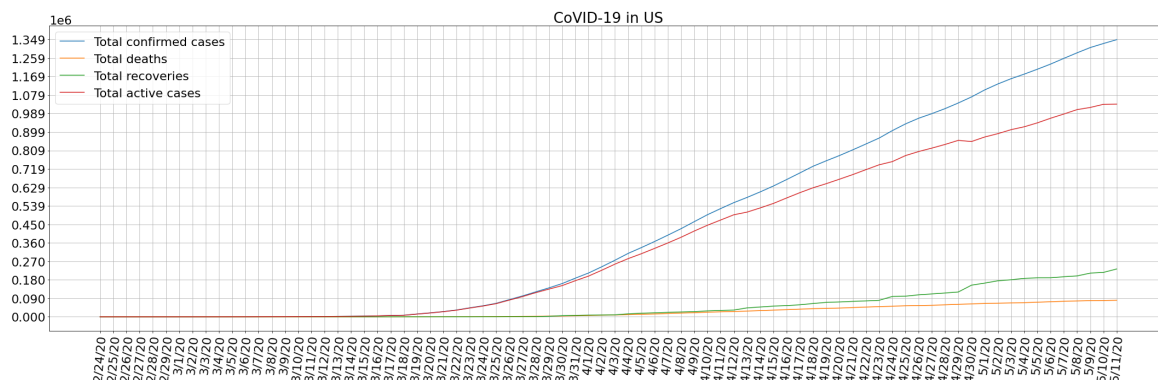
In [2]:

```
plotForCountry('the world', world_cases, world_deaths, world_recoveries)
```



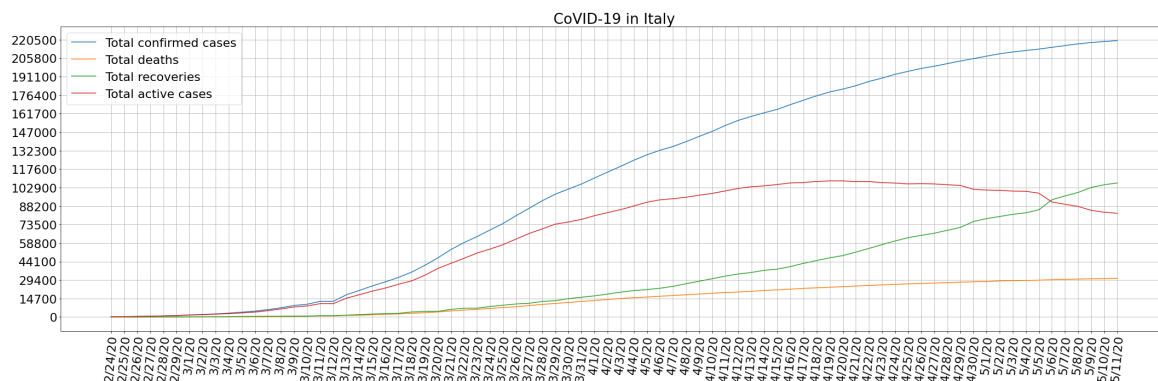
In [3]:

```
plotForCountry(country_00, country_00_cases, country_00_deaths, country_00_recoveries)
```



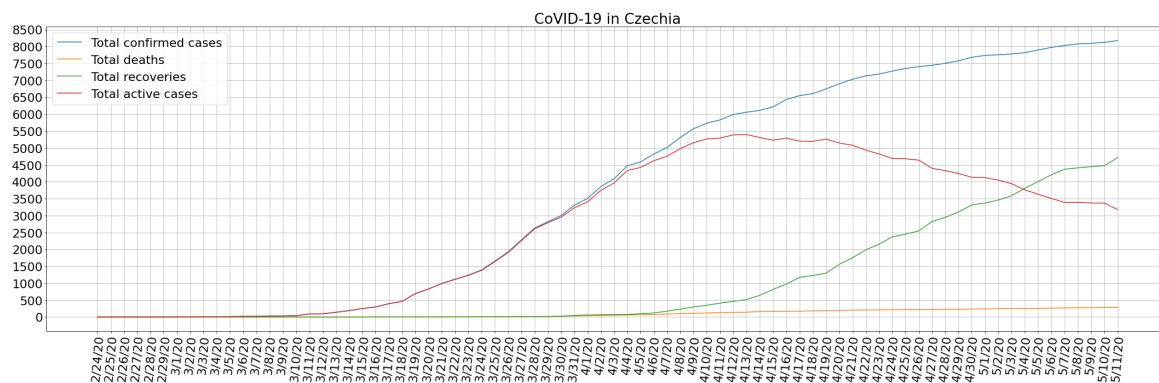
In [4]:

```
plotForCountry(country_01, country_01_cases, country_01_deaths, country_01_recoveries)
```



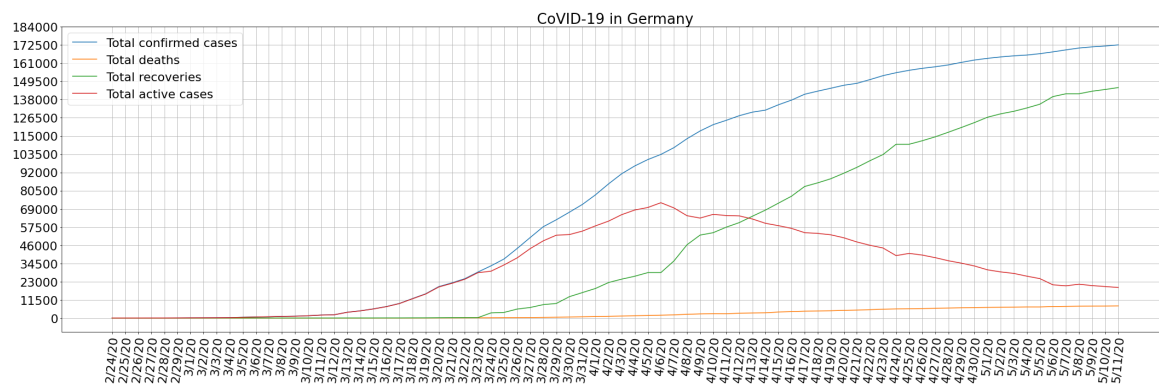
In [5]:

```
plotForCountry(country_02, country_02_cases, country_02_deaths, country_02_recoveri
```



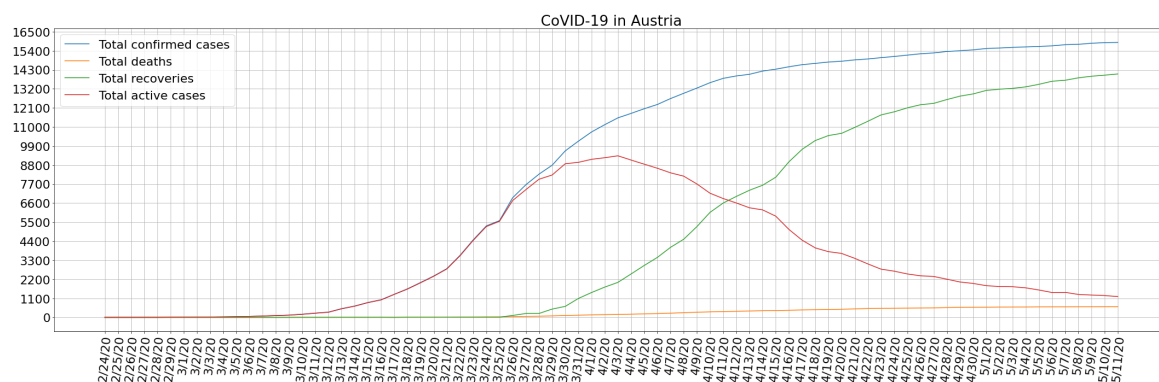
In [6]:

```
plotForCountry(country_03, country_03_cases, country_03_deaths, country_03_recoveri
```



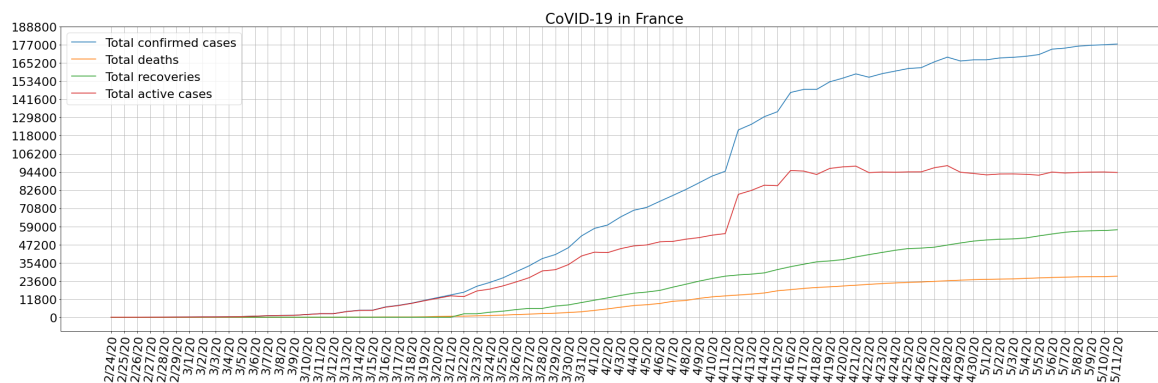
In [7]:

```
plotForCountry(country_04, country_04_cases, country_04_deaths, country_04_recoveri
```



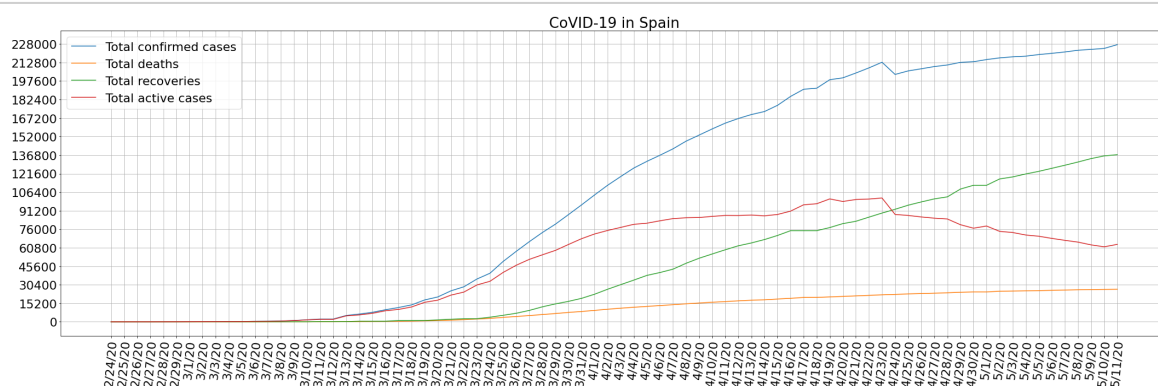
In [8]:

```
plotForCountry(country_05, country_05_cases, country_05_deaths, country_05_recoveri
```



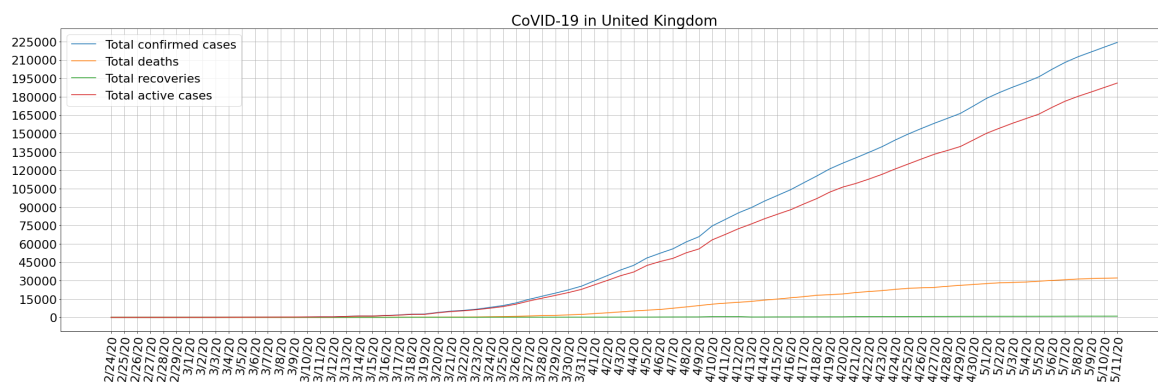
In [9]:

```
plotForCountry(country_06, country_06_cases, country_06_deaths, country_06_recoveri
```



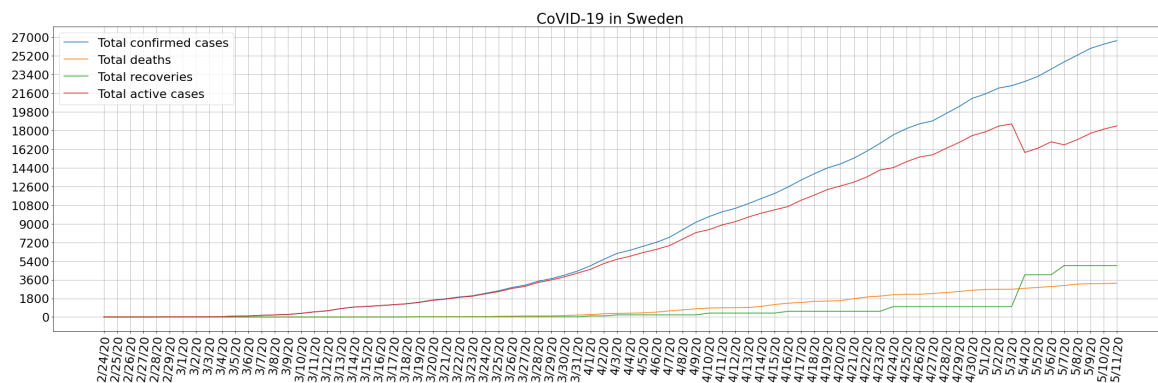
In [10]:

```
plotForCountry(country_07, country_07_cases, country_07_deaths, country_07_recoveri
```



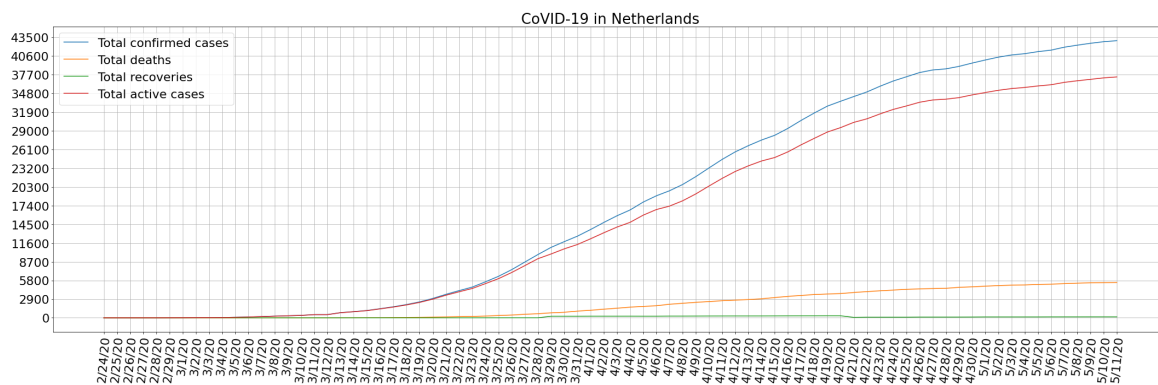
In [11]:

```
plotForCountry(country_08, country_08_cases, country_08_deaths, country_08_recoveri
```



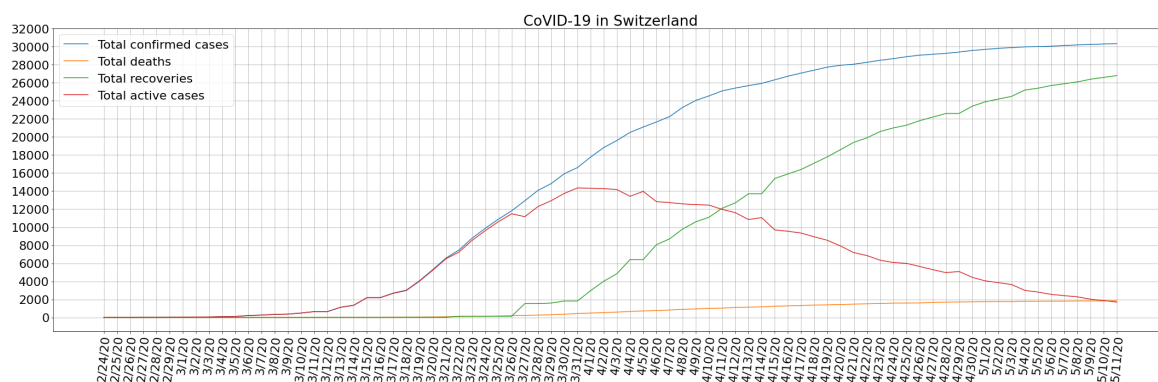
In [12]:

```
plotForCountry(country_09, country_09_cases, country_09_deaths, country_09_recoveri
```



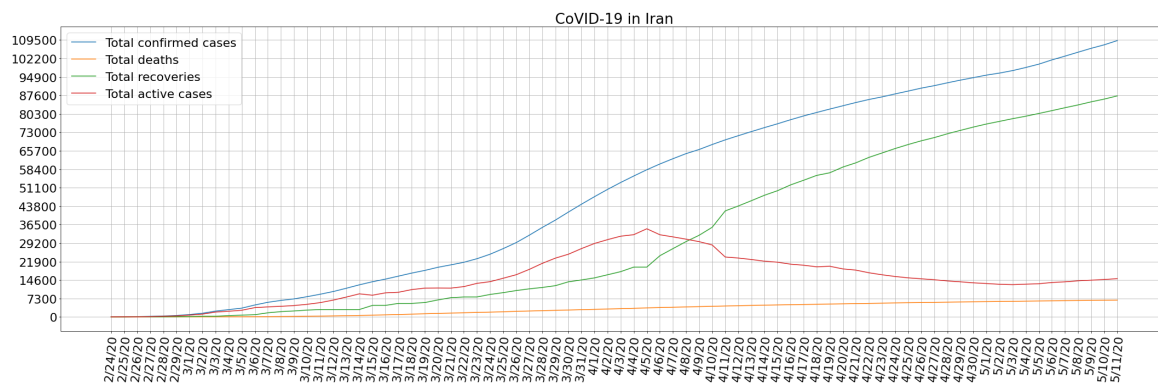
In [13]:

```
plotForCountry(country_10, country_10_cases, country_10_deaths, country_10_recoveri
```



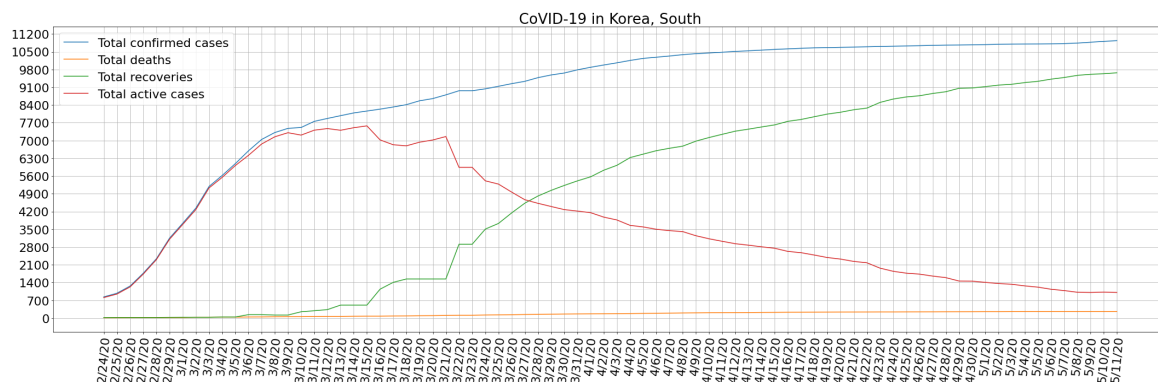
In [14]:

```
plotForCountry(country_11, country_11_cases, country_11_deaths, country_11_recoveri
```



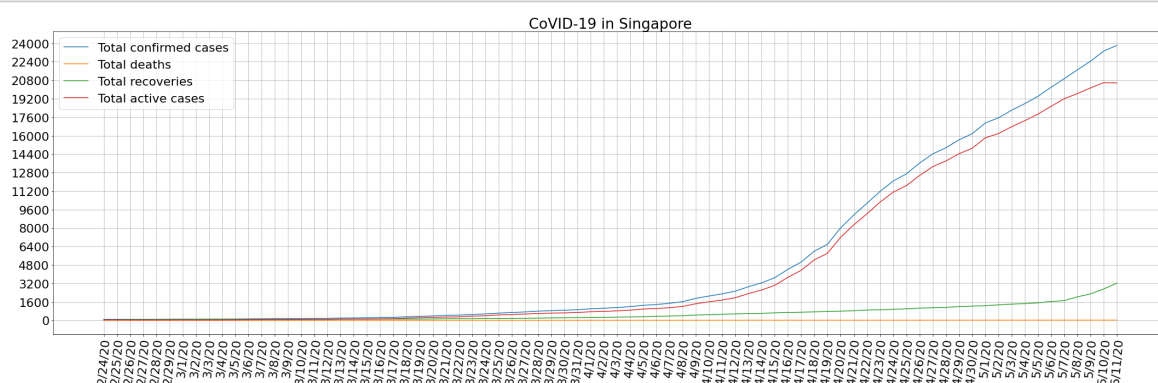
In [15]:

```
plotForCountry(country_12, country_12_cases, country_12_deaths, country_12_recoveri
```



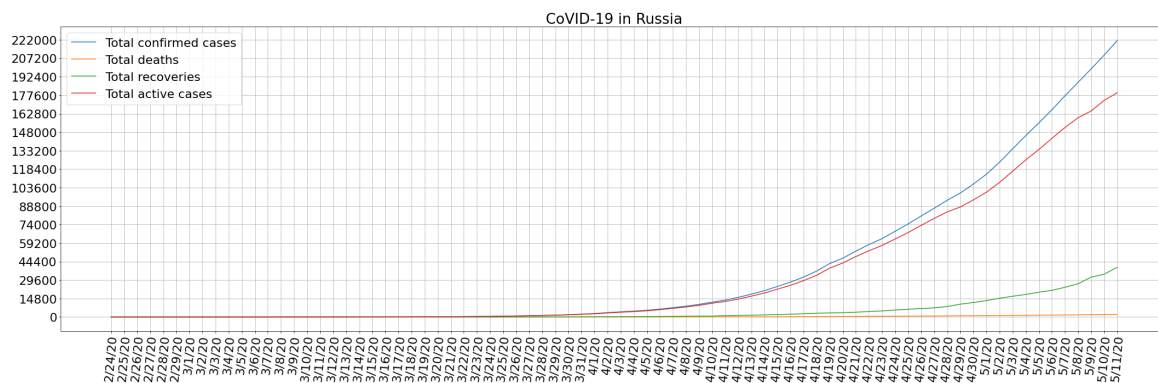
In [16]:

```
plotForCountry(country_13, country_13_cases, country_13_deaths, country_13_recoveri
```



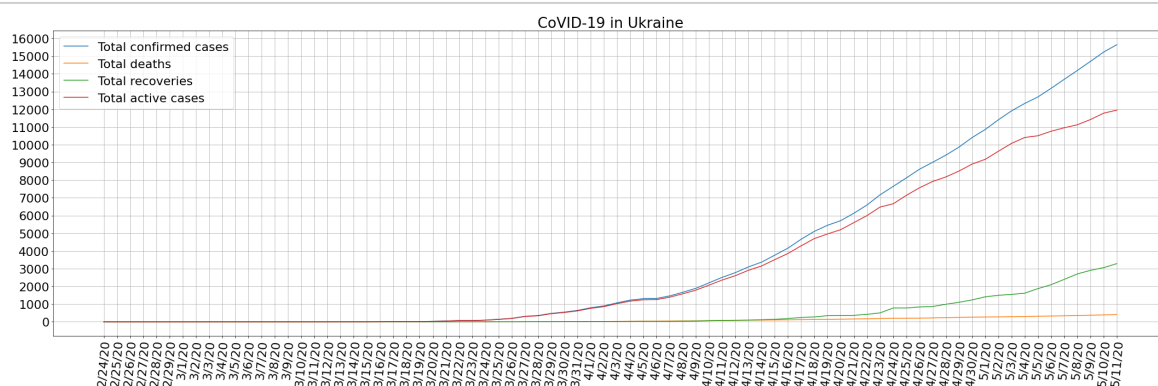
In [17]:

```
plotForCountry(country_14, country_14_cases, country_14_deaths, country_14_recoveri
```



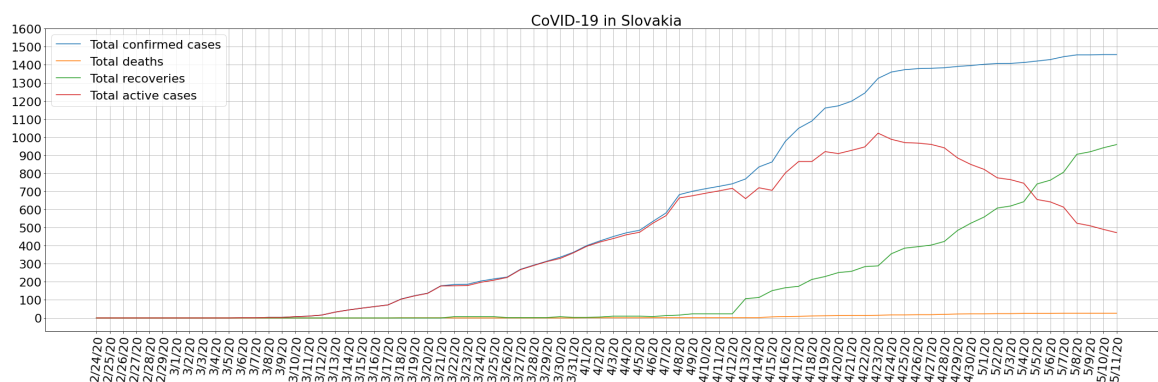
In [18]:

```
plotForCountry(country_15, country_15_cases, country_15_deaths, country_15_recoveri
```



In [19]:

```
plotForCountry(country_16, country_16_cases, country_16_deaths, country_16_recoveri
```



In [20]:

```
plotForCountry(country_17, country_17_cases, country_17_deaths, country_17_recoveri
```

