In [45]:

```python
#!pip install lmfit
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
#from scipy.optimize import curve_fit
```

In [46]:

```
data =pd.read_csv("C:/MLCourse/table.csv",sep=',')
data
```

Out[46]:

	Timestamp	Country_Region	continent	Last_Update	Confirmed	Deaths
0	0	Afghanistan	Asia	2020-01-22 00:00:00	0	0
1	1	Afghanistan	Asia	2020-01-23 00:00:00	0	0
2	2	Afghanistan	Asia	2020-01-24 00:00:00	0	0
3	3	Afghanistan	Asia	2020-01-25 00:00:00	0	0
4	4	Afghanistan	Asia	2020-01-26 00:00:00	0	0
5	5	Afghanistan	Asia	2020-01-27 00:00:00	0	0
6	6	Afghanistan	Asia	2020-01-28 00:00:00	0	0
7	7	Afghanistan	Asia	2020-01-29 00:00:00	0	0
8	8	Afghanistan	Asia	2020-01-30 00:00:00	0	0
9	9	Afghanistan	Asia	2020-01-31 00:00:00	0	0
10	10	Afghanistan	Asia	2020-02-01 00:00:00	0	0
11	11	Afghanistan	Asia	2020-02-02 00:00:00	0	0
12	12	Afghanistan	Asia	2020-02-03 00:00:00	0	0
13	13	Afghanistan	Asia	2020-02-04 00:00:00	0	0
14	14	Afghanistan	Asia	2020-02-05 00:00:00	0	0
15	15	Afghanistan	Asia	2020-02-06 00:00:00	0	0
16	16	Afghanistan	Asia	2020-02-07 00:00:00	0	0
17	17	Afghanistan	Asia	2020-02-08 00:00:00	0	0
18	18	Afghanistan	Asia	2020-02-09 00:00:00	0	0
19	19	Afghanistan	Asia	2020-02-10 00:00:00	0	0
20	20	Afghanistan	Asia	2020-02-11 00:00:00	0	0
21	21	Afghanistan	Asia	2020-02-12 00:00:00	0	0
22	22	Afghanistan	Asia	2020-02-13 00:00:00	0	0
23	23	Afghanistan	Asia	2020-02-14 00:00:00	0	0
24	24	Afghanistan	Asia	2020-02-15 00:00:00	0	0
25	25	Afghanistan	Asia	2020-02-16 00:00:00	0	0
26	26	Afghanistan	Asia	2020-02-17 00:00:00	0	0
27	27	Afghanistan	Asia	2020-02-18 00:00:00	0	0
28	28	Afghanistan	Asia	2020-02-19 00:00:00	0	0
29	29	Afghanistan	Asia	2020-02-20 00:00:00	0	0
...	...	...	...	...	...	...
13029	13547	Singapore	Asia	2020-04-05 00:00:00	1309	6
13030	13548	Slovakia	Europe	2020-04-05 00:00:00	485	1
13031	13549	Slovenia	Europe	2020-04-05 00:00:00	997	28
13032	13550	Somalia	Africa	2020-04-05 00:00:00	7	0

	Timestamp	Country_Region	continent	Last_Update	Confirmed	Deaths
13033	13551	South Africa	Africa	2020-04-05 00:00:00	1585	9
13034	13552	South Sudan	Africa	2020-04-05 00:00:00	1	0
13035	13553	Spain	Europe	2020-04-05 00:00:00	130759	12418
13036	13554	Sri Lanka	Asia	2020-04-05 00:00:00	175	5
13037	13555	Sudan	Africa	2020-04-05 00:00:00	12	2
13038	13556	Suriname	South America	2020-04-05 00:00:00	10	1
13039	13557	Syria	Asia	2020-04-05 00:00:00	19	2
13040	13558	Taiwan	Asia	2020-04-05 00:00:00	363	5
13041	13559	Tanzania	Africa	2020-04-05 00:00:00	20	1
13042	13560	Thailand	Asia	2020-04-05 00:00:00	2169	23
13043	13561	Timor-Leste	Others	2020-04-05 00:00:00	1	0
13044	13562	Togo	Africa	2020-04-05 00:00:00	44	3
13045	13563	Trinidad and Tobago	North America	2020-04-05 00:00:00	104	7
13046	13564	Tunisia	Africa	2020-04-05 00:00:00	553	19
13047	13565	Turkey	Asia	2020-04-05 00:00:00	27069	574
13048	13566	Uganda	Africa	2020-04-05 00:00:00	48	0
13049	13567	Ukraine	Europe	2020-04-05 00:00:00	1251	32
13050	13568	United Arab Emirates	Asia	2020-04-05 00:00:00	1505	10
13051	13569	Uruguay	South America	2020-04-05 00:00:00	400	5
13052	13570	Uzbekistan	Asia	2020-04-05 00:00:00	298	2
13053	13571	Venezuela	South America	2020-04-05 00:00:00	155	7
13054	13572	Vietnam	Asia	2020-04-05 00:00:00	241	0
13055	13573	West Bank and Gaza	Others	2020-04-05 00:00:00	228	1
13056	13574	Western Sahara	Others	2020-04-05 00:00:00	4	0
13057	13575	Zambia	Africa	2020-04-05 00:00:00	39	1
13058	13576	Zimbabwe	Africa	2020-04-05 00:00:00	9	1

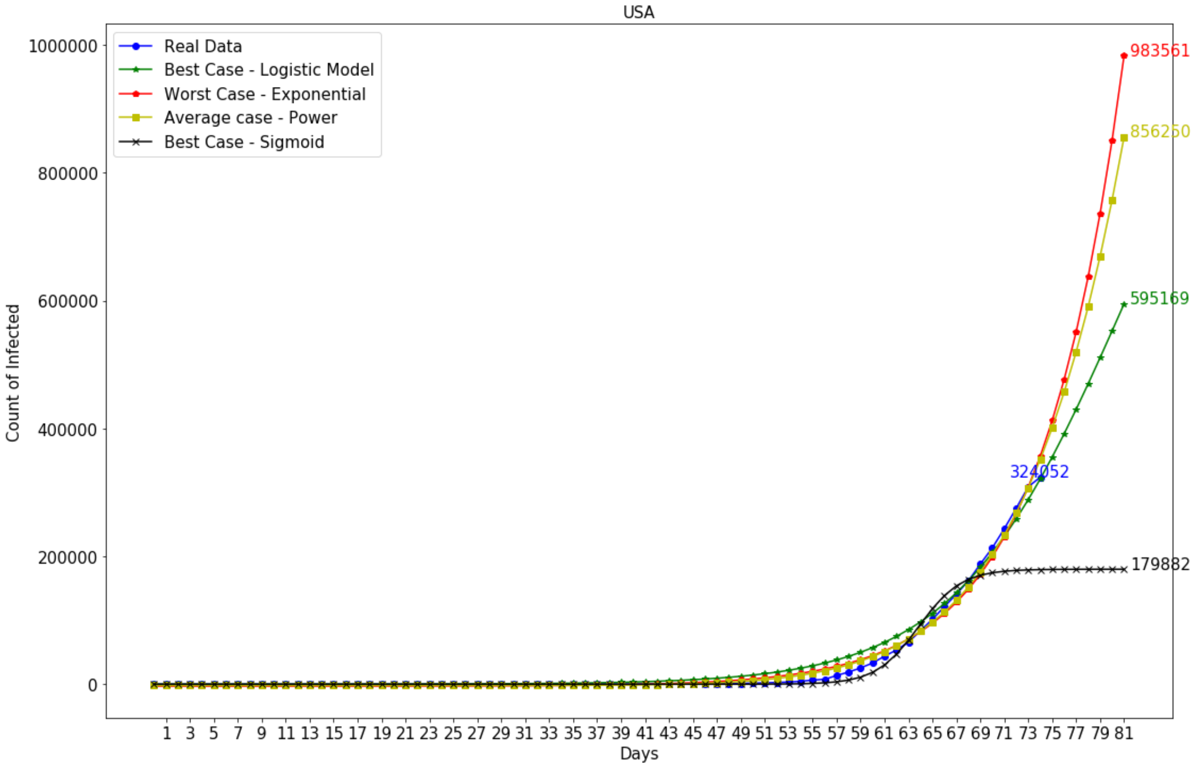
13059 rows × 6 columns

In [53]:

```
def COVID_Spread_Prediction(predicted_days, Country_Region):
    cmd = data[data["Country_Region"]==Country_Region].iloc[: , [0, 2, 3 ,4, 5
]].copy()
    cmd_grp = cmd.groupby("Last_Update")[['Confirmed', 'Deaths']].sum().reset_
index()
    y = cmd_grp["Confirmed"]
    x = np.arange(len(y))

    def power(x, a, b, c):
        return b*(x)**a + c

    def exp(x, a, b, c):
        return a * np.exp(-b * x) + c

    def sigmoid(x, a, b, c, d):
        return c / (1 + np.exp(-b*(x-a)))+d

    def logis(x,a,b,c):
        return c/(1+a*np.exp(-b*x))
    p0=np.random.exponential(size=3)
    p0
    bounds_log=(0,[100000.,3.,1000000000.])
    popt_pow, pcov_pow = curve_fit(power, x, y,maxfev=100000)
    popt_exp, pcov_exp = curve_fit(exp, x, y, p0=(1, 1e-6, 1), maxfev=100000)
    popt_sig, pcov_sig = curve_fit(sigmoid,x, y, method='dogbox', bounds=([10.
, 0.001, y.mean(), 10],[100, 1., 10*y.mean(), 100]), maxfev=200000)
    popt_log, pcov_log = curve_fit(logis,x,y,bounds=bounds_log,p0=p0)

    # Real Data
    plt.figure(figsize=(18,12))
    x1 = np.arange(len(y)+predicted_days)
    y = y.values
    plt.plot(x, y, c='b', marker="o", label = "Real Data")
    plt.text(x[-1]-2.5, y[-1], str(int(y[-1])), size = 15, color="b")

    #Logistic Growth Model
    y1 = logis(x1, *popt_log)
    plt.plot(x1, y1, c='g', marker="*", label="Best Case - Logistic Model")
    plt.text(x1[-1]+.5, y1[-1], str(int(y1[-1])), size = 15, color="g")

    #Exponential Growth Model
    y1 = exp(x1, *popt_exp)
    plt.plot(x1, y1, c='r', marker="p", label="Worst Case - Exponential")
    plt.text(x1[-1]+.5, y1[-1], str(int(y1[-1])), size = 15, color="r")

    #Power Law Model
    y1 = power(x1, *popt_pow)
    plt.plot(x1, y1, c='y', marker="s", label="Average case - Power")
    plt.text(x1[-1]+.5, y1[-1], str(int(y1[-1])), size = 15, color="y")
    #Sigmoid Function Model
    y1 = sigmoid(x1, *popt_sig)
    plt.plot(x1, y1, c='k', marker="x", label="Best Case - Sigmoid")
    plt.text(x1[-1]+.5, y1[-1], str(int(y1[-1])), size = 15, color="k")
    plt.xlabel("Days", size=15)
    plt.xticks(np.arange(1,len(x1),2),size=15)
    plt.ylabel("Count of Infected", size=15)
    plt.yticks(size=15)
    plt.legend(prop={'size': 15})
    plt.title(Country_Region, size=15)
    plt.show()

#from lmfit.models import ParabolicModel
#qmodel = ParabolicModel()
#result = qmodel.fit(y, x=x, a=1, b=2, c=0)
#print(result.fit_report())
```

In [56]:

```
predicted_days = 7
Country_Region = "USA"
COVID_Spread_Prediction(predicted_days, Country_Region)
```


USA
Real Data
Best Case - Logistic Model
Worst Case - Exponential
Average case - Power
Best Case - Sigmoid
Count of Infected
Days