In [2]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
df=pd.read_csv(r"C:\Users\Lenovo\OneDrive\Desktop\covid_19_india.csv")
df
```

Out[2]:

	Sno	Date	Time	State/UnionTerritory	ConfirmedIndianNational	ConfirmedForeignNatio
0	1.0	30/01/20	6:00 PM	Kerala	1	
1	2.0	31/01/20	6:00 PM	Kerala	1	
2	3.0	01/02/20	6:00 PM	Kerala	2	
3	4.0	02/02/20	6:00 PM	Kerala	3	
4	5.0	03/02/20	6:00 PM	Kerala	3	
...	...	...	...	...	...	
4597	4598.0	28/07/20	8:00 AM	Tripura	-	
4598	4599.0	28/07/20	8:00 AM	Uttarakhand	-	
4599	4600.0	28/07/20	8:00 AM	Uttar Pradesh	-	
4600	4601.0	28/07/20	8:00 AM	West Bengal	-	
4601	NaN	NaN	NaN	NaN	NaN	N

4602 rows × 10 columns

In [3]:
```
df1=df.drop(['Time','ConfirmedIndianNational', 'ConfirmedForeignNationa
l','Unnamed: 9'], axis=1)
```

In [4]:
```
active=df.Confirmed-(df.Cured+df.Deaths)
df1['Active']=active
df1.head()
```

Out[4]:

	Sno	Date	State/UnionTerritory	Cured	Deaths	Confirmed	Active
0	1.0	30/01/20	Kerala	0.0	0.0	1.0	1.0
1	2.0	31/01/20	Kerala	0.0	0.0	1.0	1.0
2	3.0	01/02/20	Kerala	0.0	0.0	2.0	2.0
3	4.0	02/02/20	Kerala	0.0	0.0	3.0	3.0
4	5.0	03/02/20	Kerala	0.0	0.0	3.0	3.0

In [5]:
```
df1.sort_values(["Active","Date"], axis=0,ascending=True, inplace=True)
df1
```

Out[5]:

	Sno	Date	State/UnionTerritory	Cured	Deaths	Confirmed	Active
1480	1481.0	01/05/20	Arunachal Pradesh	1.0	0.0	1.0	0.0
1486	1487.0	01/05/20	Goa	7.0	0.0	7.0	0.0
1497	1498.0	01/05/20	Manipur	2.0	0.0	2.0	0.0
1506	1507.0	01/05/20	Tripura	2.0	0.0	2.0	0.0
2522	2523.0	01/06/20	Andaman and Nicobar Islands	33.0	0.0	33.0	0.0
...	...	...	...	...	...	...	...
4480	4481.0	25/07/20	Maharashtra	199967.0	13132.0	357117.0	144018.0
4515	4516.0	26/07/20	Maharashtra	207194.0	13389.0	366368.0	145785.0
4585	4586.0	28/07/20	Maharashtra	221944.0	13883.0	383723.0	147896.0
4550	4551.0	27/07/20	Maharashtra	213238.0	13656.0	375799.0	148905.0

	Sno	Date	State/UnionTerritory	Cured	Deaths	Confirmed	Active
4601	NaN	NaN	NaN	NaN	NaN	NaN	NaN

4602 rows × 7 columns

In [6]:
```
print("STATE WITH MAXIMUM NUMBER OF CASES")
p=df1['Active'].max()
print(p)
print(df1.iloc[-2]['State/UnionTerritory'])
print("\n")
print("STATE WITH MINIMUM NUMBER OF CASES")
q=df1['Active'].min()
print(q)
print(df1.iloc[0]['State/UnionTerritory'])
```

```
STATE WITH MAXIMUM NUMBER OF CASES
148905.0
Maharashtra


STATE WITH MINIMUM NUMBER OF CASES
0.0
Arunachal Pradesh
```

In [7]:
```
df1['Ratio'] = df1['Deaths']/df1['Cured'].sum()
df1
```

Out[7]:

	Sno	Date	State/UnionTerritory	Cured	Deaths	Confirmed	Active	Ratio
1480	1481.0	01/05/20	Arunachal Pradesh	1.0	0.0	1.0	0.0	0.000000
1486	1487.0	01/05/20	Goa	7.0	0.0	7.0	0.0	0.000000
1497	1498.0	01/05/20	Manipur	2.0	0.0	2.0	0.0	0.000000
1506	1507.0	01/05/20	Tripura	2.0	0.0	2.0	0.0	0.000000
2522	2523.0	01/06/20	Andaman and Nicobar Islands	33.0	0.0	33.0	0.0	0.000000

	Sno	Date	State/UnionTerritory	Cured	Deaths	Confirmed	Active	Ratio
...	...	...	...	...	...	...	...	...
4480	4481.0	25/07/20	Maharashtra	199967.0	13132.0	357117.0	144018.0	0.000552
4515	4516.0	26/07/20	Maharashtra	207194.0	13389.0	366368.0	145785.0	0.000563
4585	4586.0	28/07/20	Maharashtra	221944.0	13883.0	383723.0	147896.0	0.000584
4550	4551.0	27/07/20	Maharashtra	213238.0	13656.0	375799.0	148905.0	0.000574
4601	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

4602 rows × 8 columns

In [8]:
```
df1.sort_values(["Ratio","Date"], axis=0,ascending=True, inplace=True)
df1
```

Out[8]:

	Sno	Date	State/UnionTerritory	Cured	Deaths	Confirmed	Active	Ratio
2	3.0	01/02/20	Kerala	0.0	0.0	2.0	2.0	0.000000
31	32.0	01/03/20	Kerala	0.0	0.0	3.0	3.0	0.000000
531	532.0	01/04/20	Assam	0.0	0.0	1.0	1.0	0.000000
541	542.0	01/04/20	Jharkhand	0.0	0.0	1.0	1.0	0.000000
547	548.0	01/04/20	Manipur	0.0	0.0	1.0	1.0	0.000000
...	...	...	...	...	...	...	...	...
4480	4481.0	25/07/20	Maharashtra	199967.0	13132.0	357117.0	144018.0	0.000552
4515	4516.0	26/07/20	Maharashtra	207194.0	13389.0	366368.0	145785.0	0.000563
4550	4551.0	27/07/20	Maharashtra	213238.0	13656.0	375799.0	148905.0	0.000574
4585	4586.0	28/07/20	Maharashtra	221944.0	13883.0	383723.0	147896.0	0.000584
4601	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

4602 rows × 8 columns

In [9]:
```
print("STATE WITH MAXIMUM DEATH TO CURE RATIO")
print(df1.iloc[-2]['State/UnionTerritory'])
print("\n")
print("STATE WITH MINIMUM DEATH TO CURE RATIO")
print(df1.iloc[0]['State/UnionTerritory'])
```

```
STATE WITH MAXIMUM DEATH TO CURE RATIO
Maharashtra


STATE WITH MINIMUM DEATH TO CURE RATIO
Kerala
```

In [13]:
```
df1.rename(columns = {'State/UnionTerritory':'State'}, inplace = True)
df1_list=['Delhi']
df1_sv=df1[df1.State.isin(df1_list)]
df1_sv
```

Out[13]:

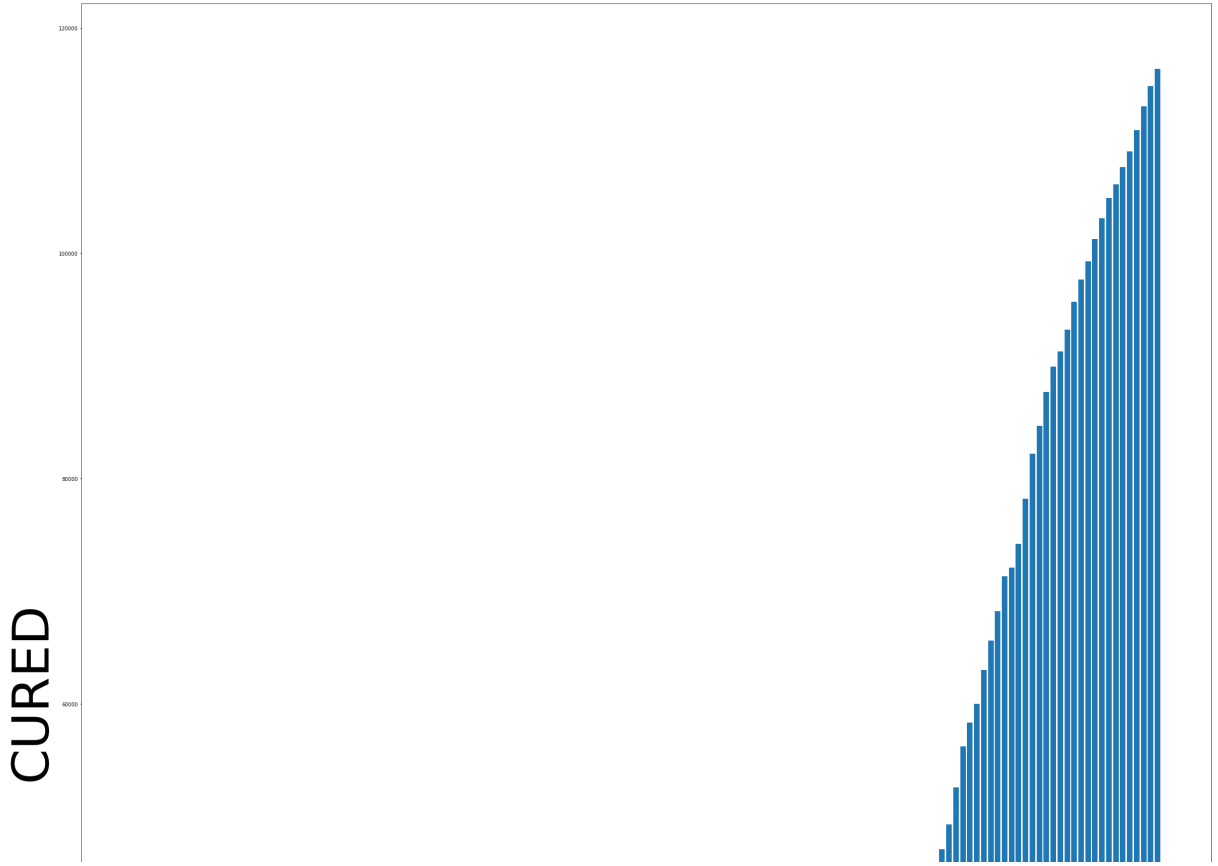
	Sno	Date	State	Cured	Deaths	Confirmed	Active	Ratio
34	35.0	02/03/20	Delhi	0.0	0.0	1.0	1.0	0.000000
38	39.0	03/03/20	Delhi	0.0	0.0	1.0	1.0	0.000000
42	43.0	04/03/20	Delhi	0.0	0.0	1.0	1.0	0.000000
45	46.0	05/03/20	Delhi	0.0	0.0	2.0	2.0	0.000000
51	52.0	06/03/20	Delhi	0.0	0.0	3.0	3.0	0.000000
...	...	...	...	...	...	...	...	...
4434	4435.0	24/07/20	Delhi	109065.0	3745.0	127364.0	14554.0	0.000157
4469	4470.0	25/07/20	Delhi	110931.0	3777.0	128389.0	13681.0	0.000159
4504	4505.0	26/07/20	Delhi	113068.0	3806.0	129531.0	12657.0	0.000160
4539	4540.0	27/07/20	Delhi	114875.0	3827.0	130606.0	11904.0	0.000161
4574	4575.0	28/07/20	Delhi	116372.0	3853.0	131219.0	10994.0	0.000162

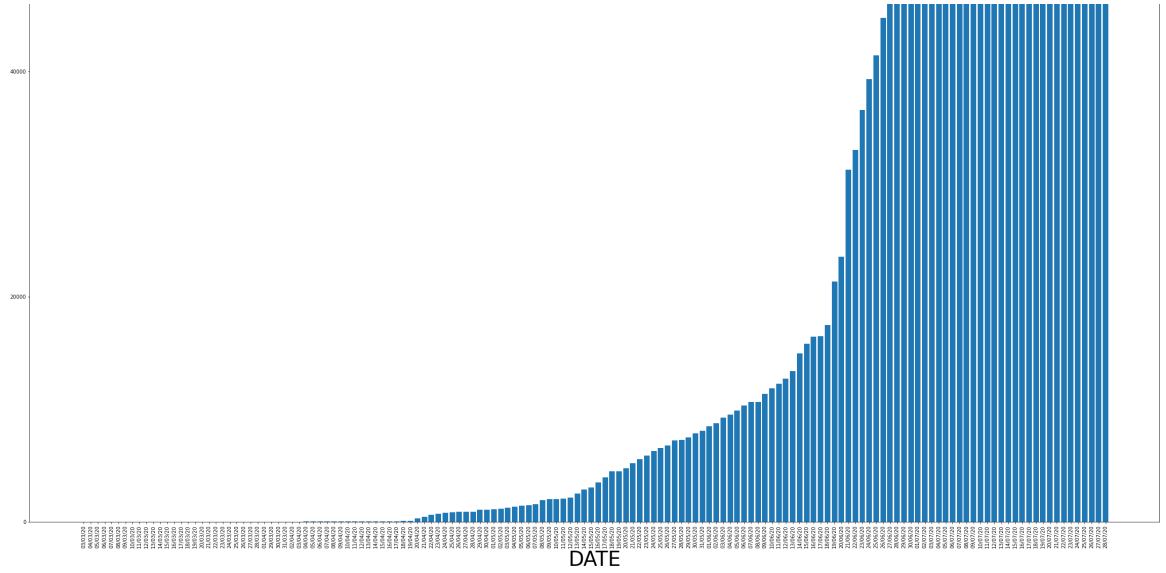
149 rows × 8 columns

In [14]:
```
print("DETAILS OF DELHI")
print("CURED vs DATE")
plt.rcParams["figure.figsize"] = (40,50)
x=df1_sv['Date'].tail(148)
y=df1_sv['Cured'].tail(148)
plt.xlabel('DATE', fontsize=40)
plt.ylabel('CURED', fontsize=100)
plt.xticks(rotation=90)
plt.bar(x,y)
```

```
DETAILS OF DELHI
CURED vs DATE
```

Out[14]:
```
<BarContainer object of 148 artists>
```

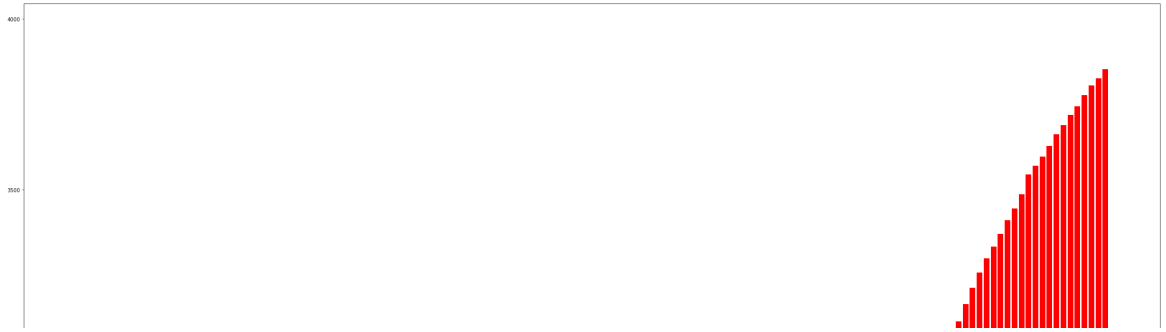


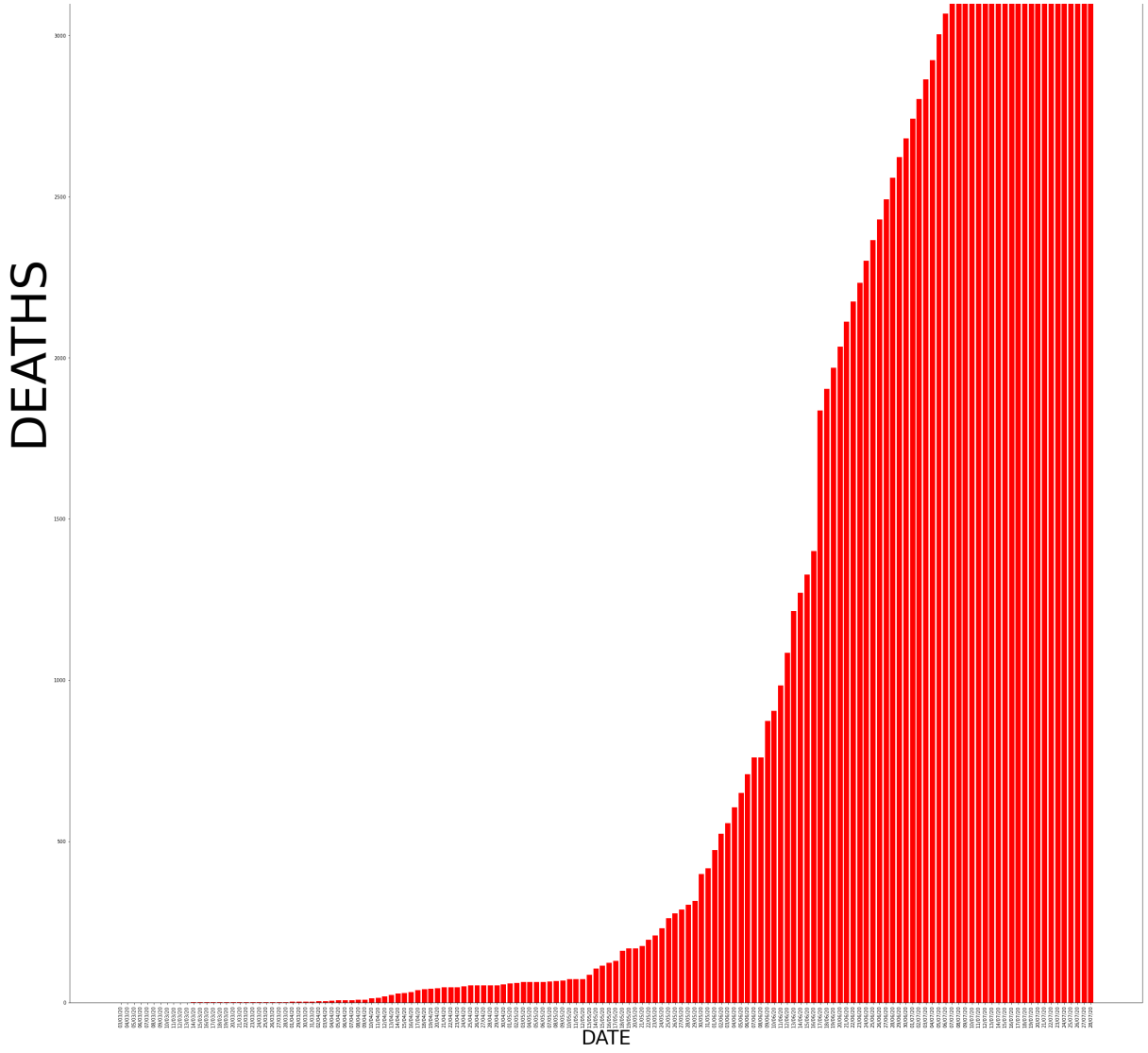


In [15]:
```
print("DEATH vs DATE")
plt.rcParams["figure.figsize"] = (40,50)
x=df1_sv['Date'].tail(148)
y=df1_sv['Deaths'].tail(148)
plt.xlabel('DATE', fontsize=40)
plt.ylabel('DEATHS', fontsize=100)
plt.xticks(rotation=90)
plt.bar(x,y,color='red')
```

```
DEATH vs DATE
```

Out[15]:
```
<BarContainer object of 148 artists>
```





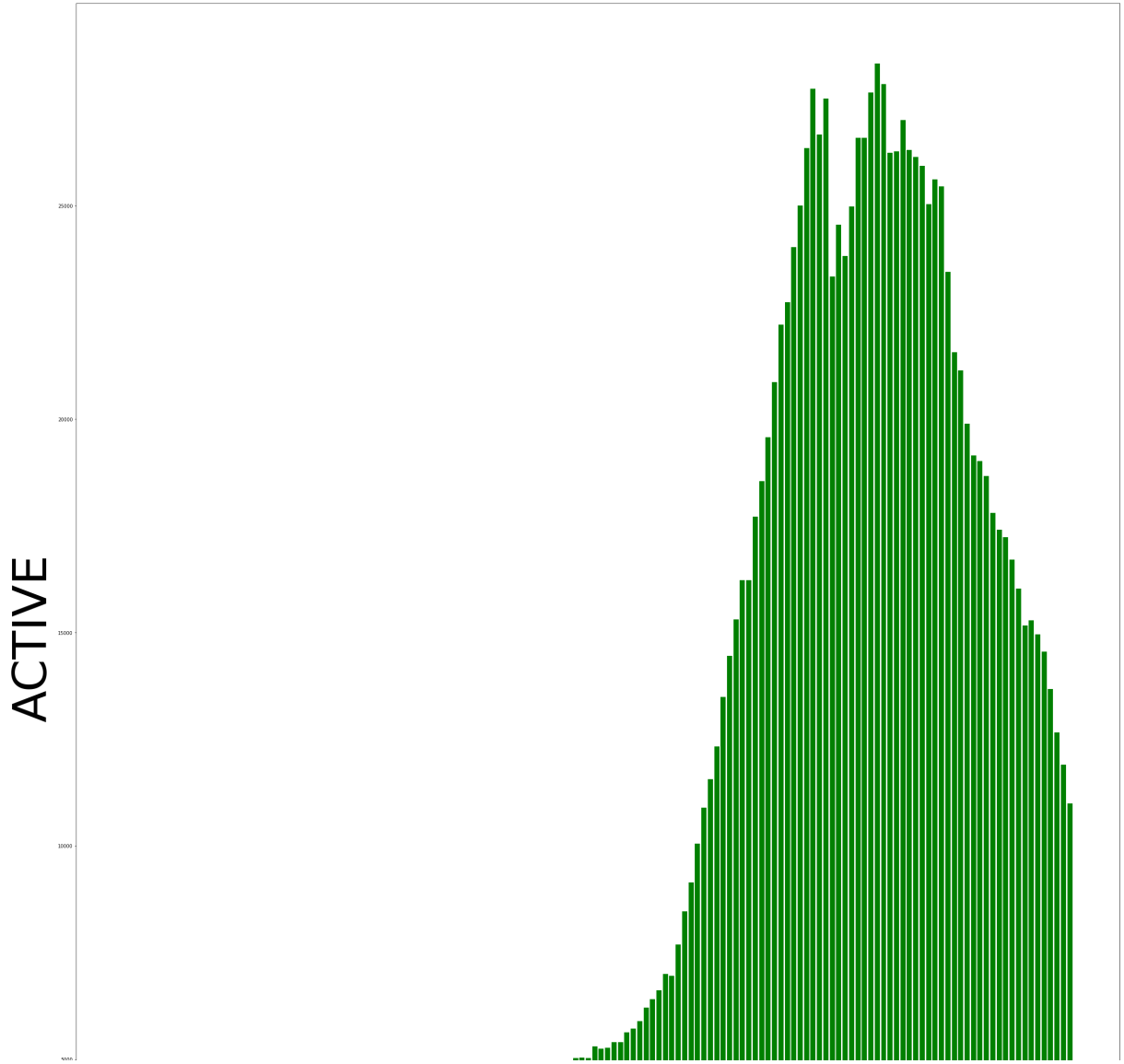
In [17]:
```
print("ACTIVE vs DATE")
plt.rcParams["figure.figsize"] = (40,50)
x=df1_sv['Date'].tail(148)
y=df1_sv['Active'].tail(148)
plt.xlabel('DATE', fontsize=40)
plt.ylabel('ACTIVE', fontsize=100)
```

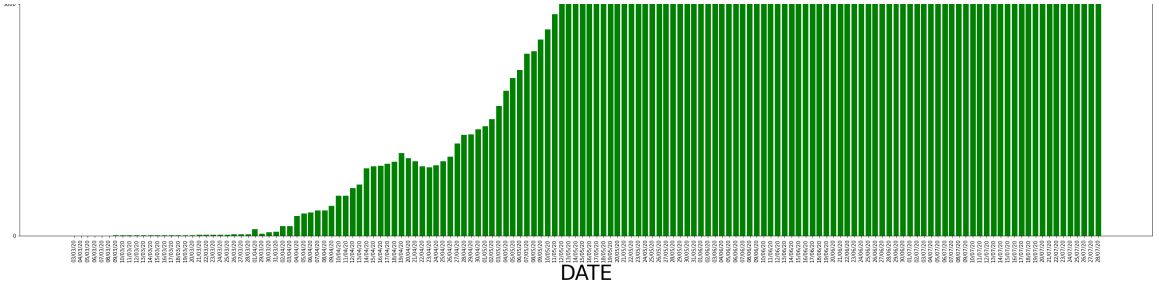
```
plt.xticks(rotation=90)
plt.bar(x,y,color='green')
```

```
ACTIVE vs DATE
```

Out[17]:

```
<BarContainer object of 148 artists>
```





In [18]:
```
df1_list=['Maharashtra']
df1_sv1=df1[df1.State.isin(df1_list)]
df1_sv1
```

Out[18]:

	Sno	Date	State	Cured	Deaths	Confirmed	Active	Ratio
76	77.0	09/03/20	Maharashtra	0.0	0.0	2.0	2.0	0.000000
91	92.0	10/03/20	Maharashtra	0.0	0.0	5.0	5.0	0.000000
97	98.0	11/03/20	Maharashtra	0.0	0.0	2.0	2.0	0.000000
120	121.0	12/03/20	Maharashtra	0.0	0.0	11.0	11.0	0.000000
133	134.0	13/03/20	Maharashtra	0.0	0.0	14.0	14.0	0.000000
...	...	...	...	...	...	...	...	...
4445	4446.0	24/07/20	Maharashtra	194253.0	12854.0	347502.0	140395.0	0.000540
4480	4481.0	25/07/20	Maharashtra	199967.0	13132.0	357117.0	144018.0	0.000552
4515	4516.0	26/07/20	Maharashtra	207194.0	13389.0	366368.0	145785.0	0.000563
4550	4551.0	27/07/20	Maharashtra	213238.0	13656.0	375799.0	148905.0	0.000574
4585	4586.0	28/07/20	Maharashtra	221944.0	13883.0	383723.0	147896.0	0.000584

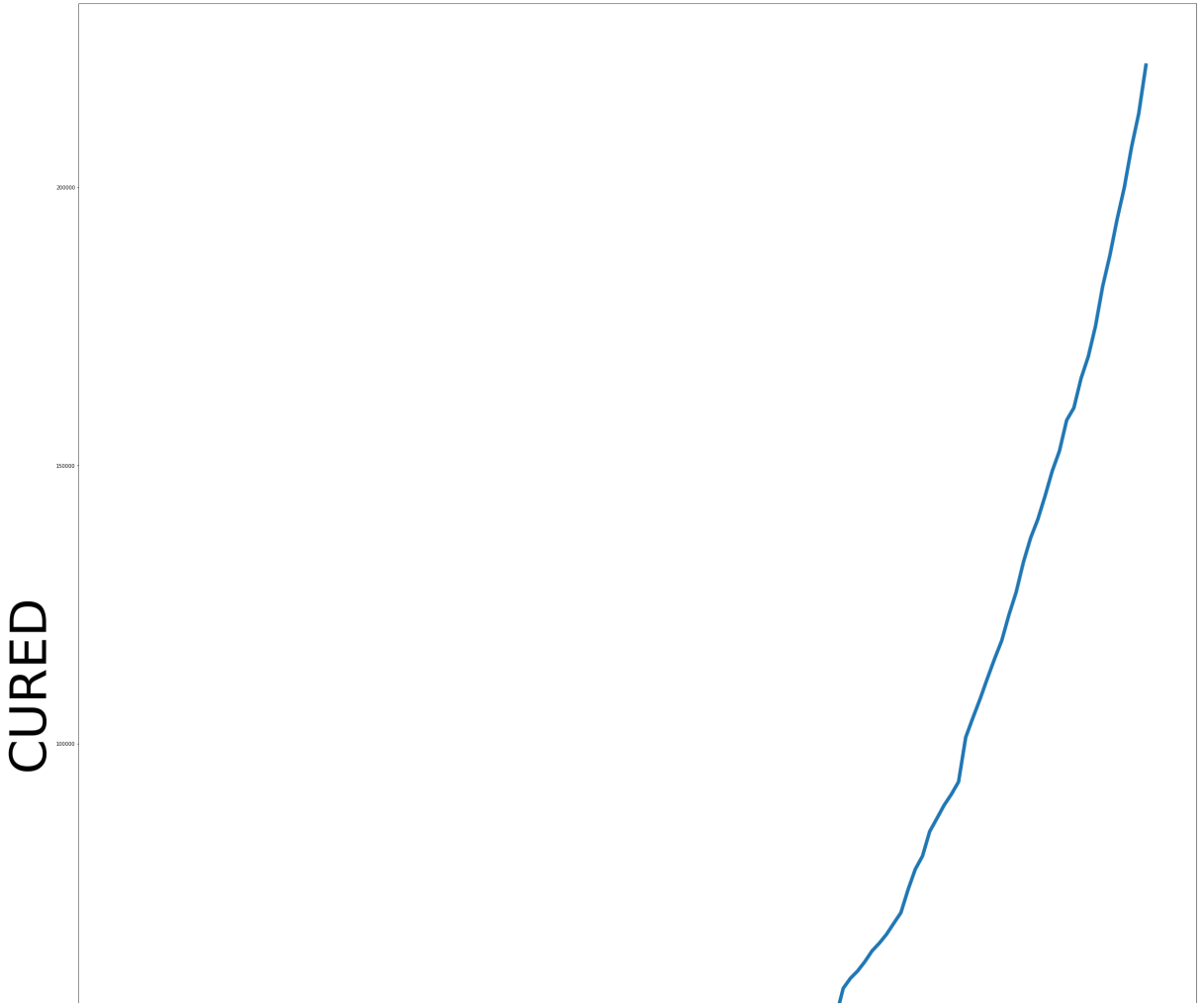
142 rows × 8 columns

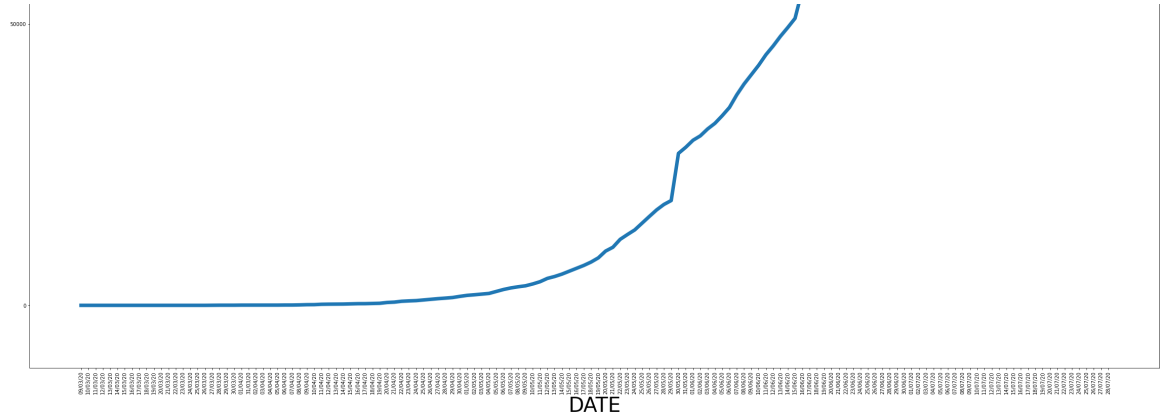
In [31]:
```
print("DETAILS OF MAHARASTRA")
print("CURED vs DATE")
plt.rcParams["figure.figsize"] = (40,50)
x=df1_sv1['Date'].tail(148)
```

```
y=df1_sv1['Cured'].tail(148)
plt.xlabel('DATE', fontsize=40)
plt.ylabel('CURED', fontsize=100)
plt.xticks(rotation=90)
plt.plot(x,y,linewidth=7,markersize=12)
```

```
DETAILS OF MAHARASTRA
CURED vs DATE
```

Out[31]: `[<matplotlib.lines.Line2D at 0x207228f18e0>]`

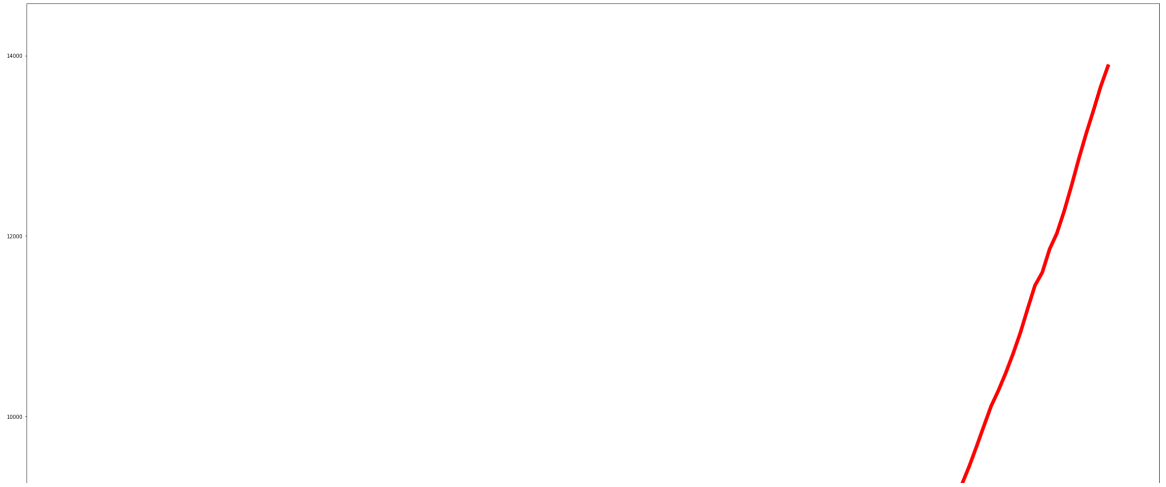


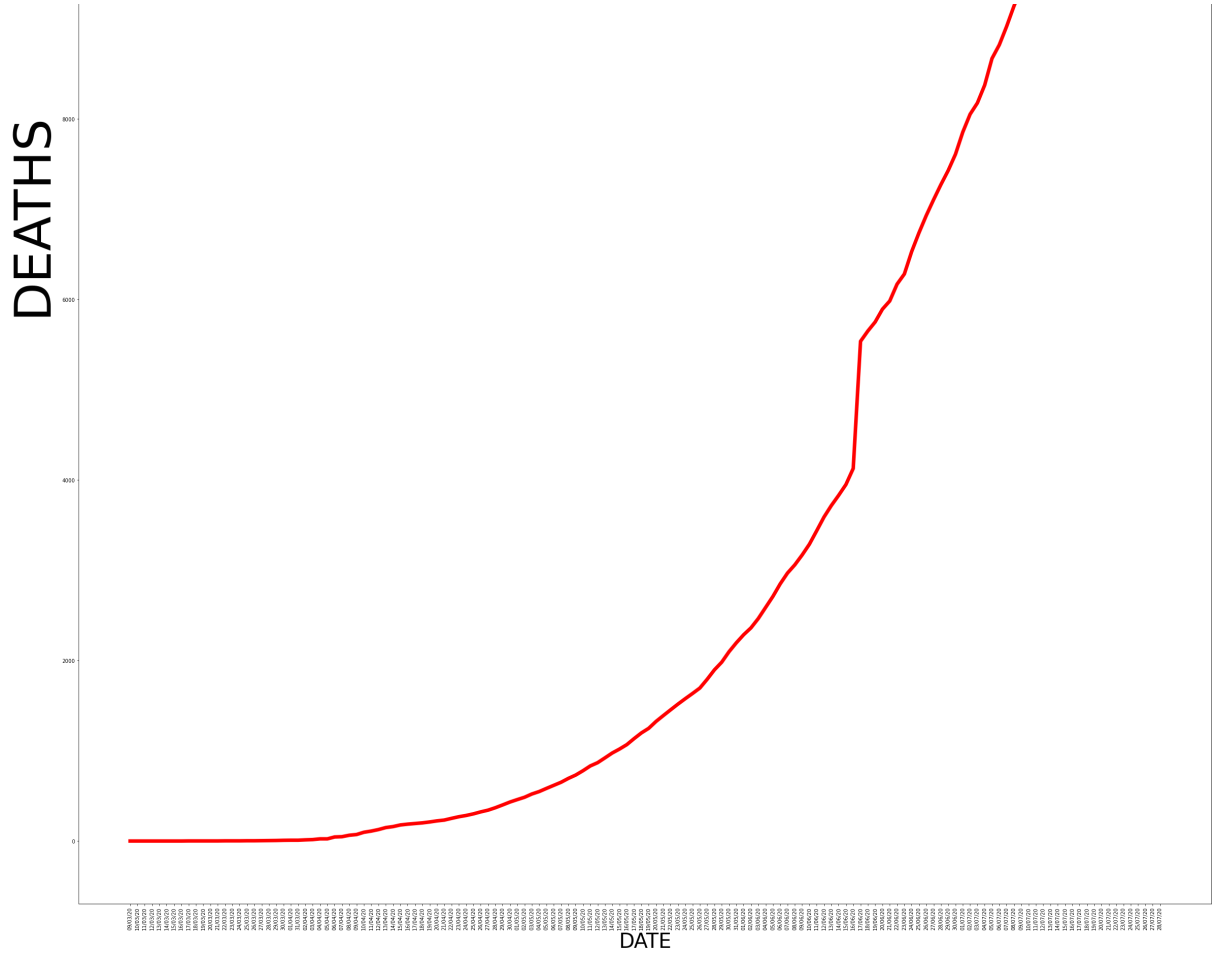


In [30]:
```
print("DEATH vs DATE")
plt.rcParams["figure.figsize"] = (40,50)
x=df1_sv1['Date'].tail(148)
y=df1_sv1['Deaths'].tail(148)
plt.xlabel('DATE', fontsize=40)
plt.ylabel('DEATHS', fontsize=100)
plt.xticks(rotation=90)
plt.plot(x,y,color='red',linewidth=7,markersize=12)
```

```
DEATH vs DATE
```

Out[30]:
```
[<matplotlib.lines.Line2D at 0x20722794ac0>]
```

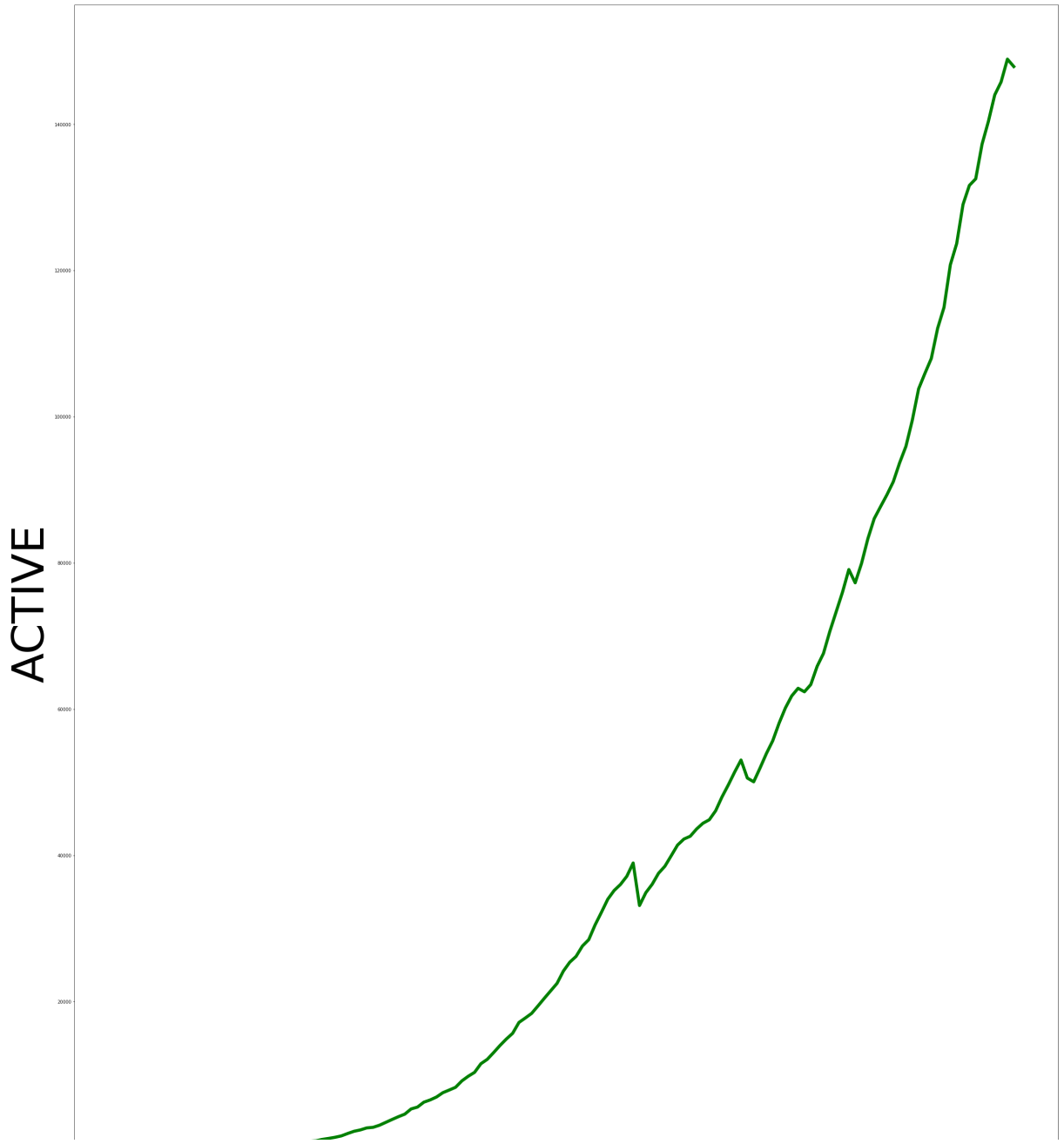


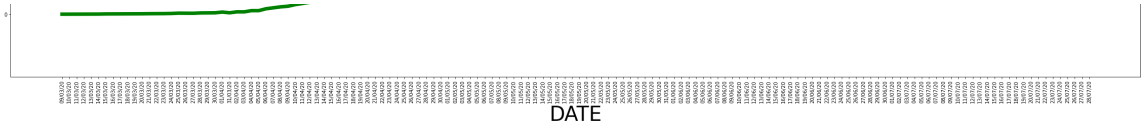


In [32]:
```
print("ACTIVE vs DATE")
plt.rcParams["figure.figsize"] = (40,50)
x=df1_sv1['Date'].head(142)
y3=df1_sv1['Active'].head(142)
plt.xlabel('DATE', fontsize=40)
plt.ylabel('ACTIVE', fontsize=100)
plt.xticks(rotation=90)
plt.plot(x,y3,color='green',linewidth=7,markersize=12)
```

```
ACTIVE vs DATE
```

Out[32]: [<matplotlib.lines.Line2D at 0x2072a243220>]

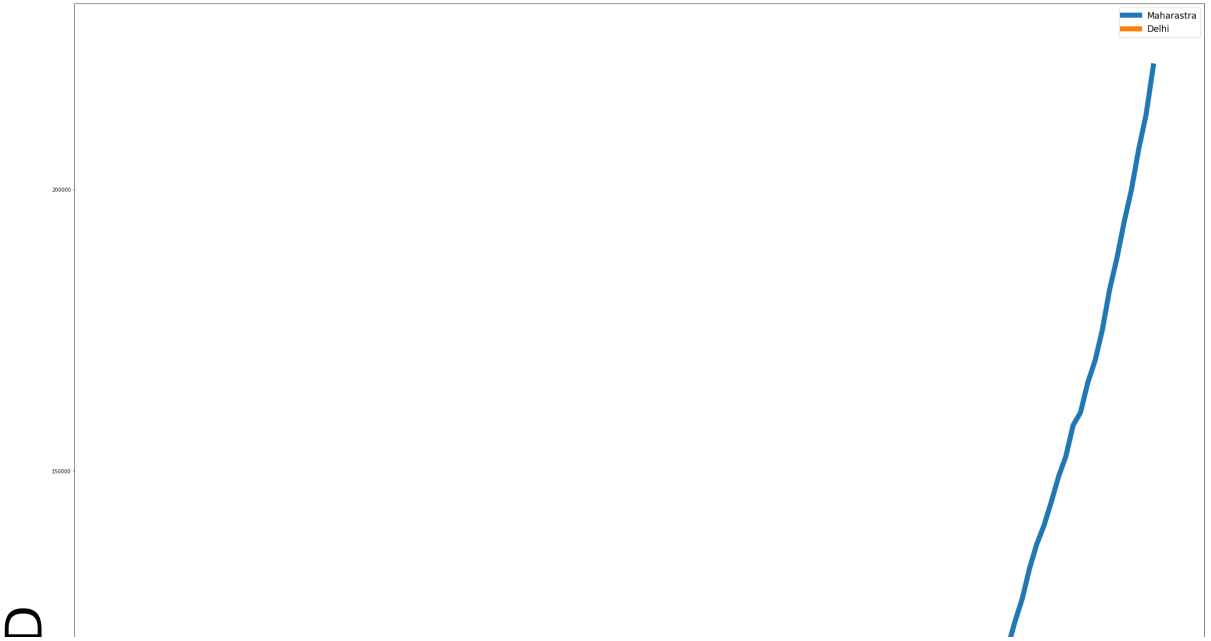


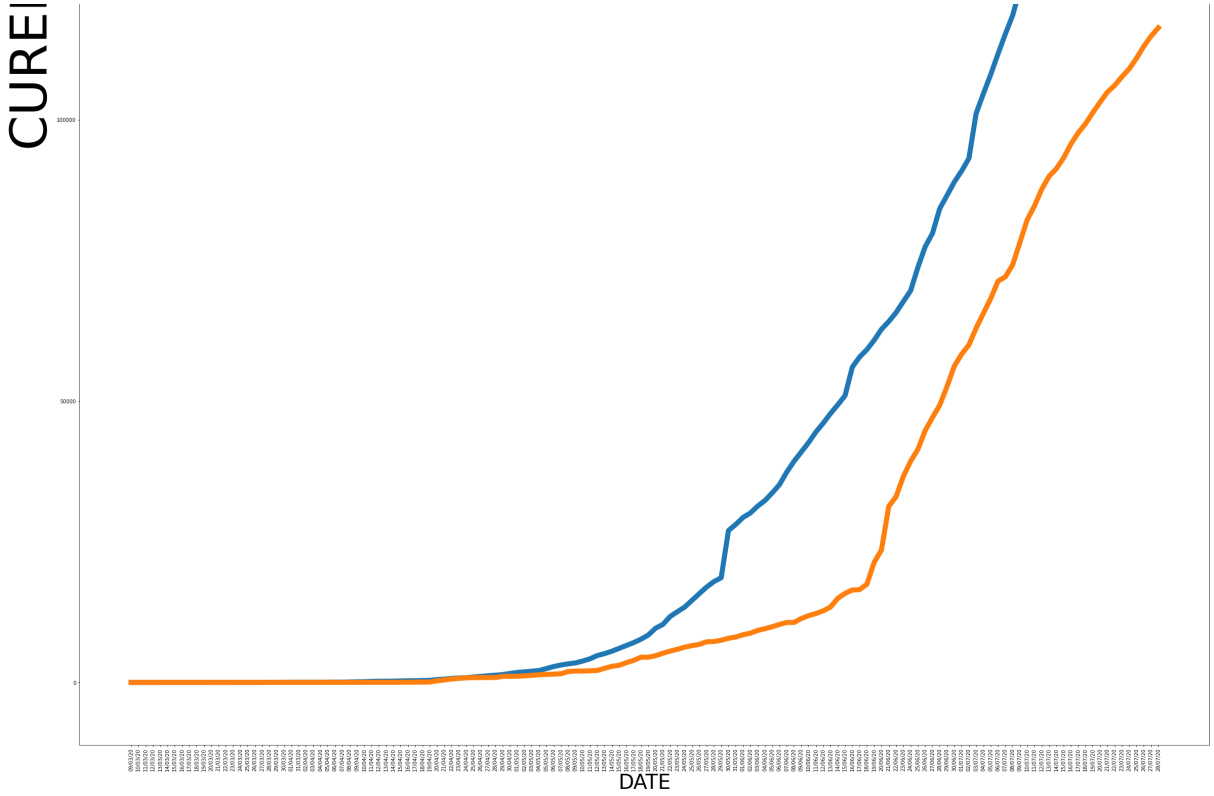


In [33]:
```
print("COMPARISION BETWEEN DELHI AND MAHARASTRA")
print("CURED vs DATE")
x=df1_sv1['Date'].tail(142)
y4=df1_sv['Cured'].tail(142)
plt.xticks(rotation=90)
y5=df1_sv1['Cured'].tail(142)
plt.xlabel('DATE', fontsize=40)
plt.ylabel('CURED', fontsize=100)
plt.rc('legend', fontsize='xx-large')
plt.plot(x,y5,x,y4,linewidth=10)
plt.legend(['Maharastra', 'Delhi'])
```

```
COMPARISION BETWEEN DELHI AND MAHARASTRA
CURED vs DATE
```

Out[33]: <matplotlib.legend.Legend at 0x20722627460>

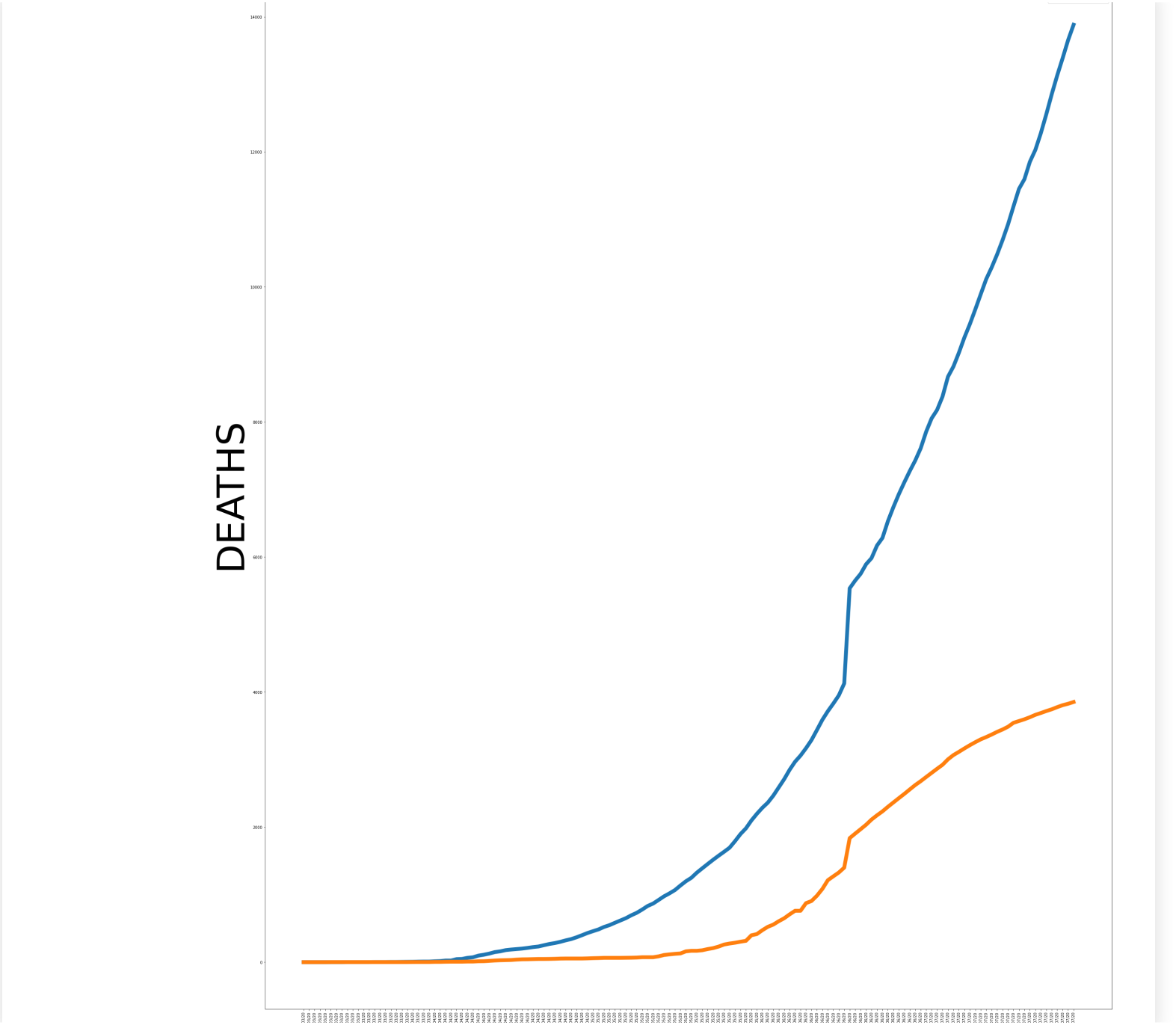




In [28]:
```
print("DEATH vs DATE")
x=df1_sv['Date'].tail(142)
y4=df1_sv['Deaths'].tail(142)
plt.xticks(rotation=90)
y5=df1_sv1['Deaths'].tail(142)
plt.xlabel('DATE', fontsize=40)
plt.ylabel('DEATHS', fontsize=100)
plt.rc('legend', fontsize='xx-large')
plt.plot(x,y5,x,y4,linewidth=10)
plt.legend(['Maharastra', 'Delhi'])
```

```
DEATH vs DATE
```

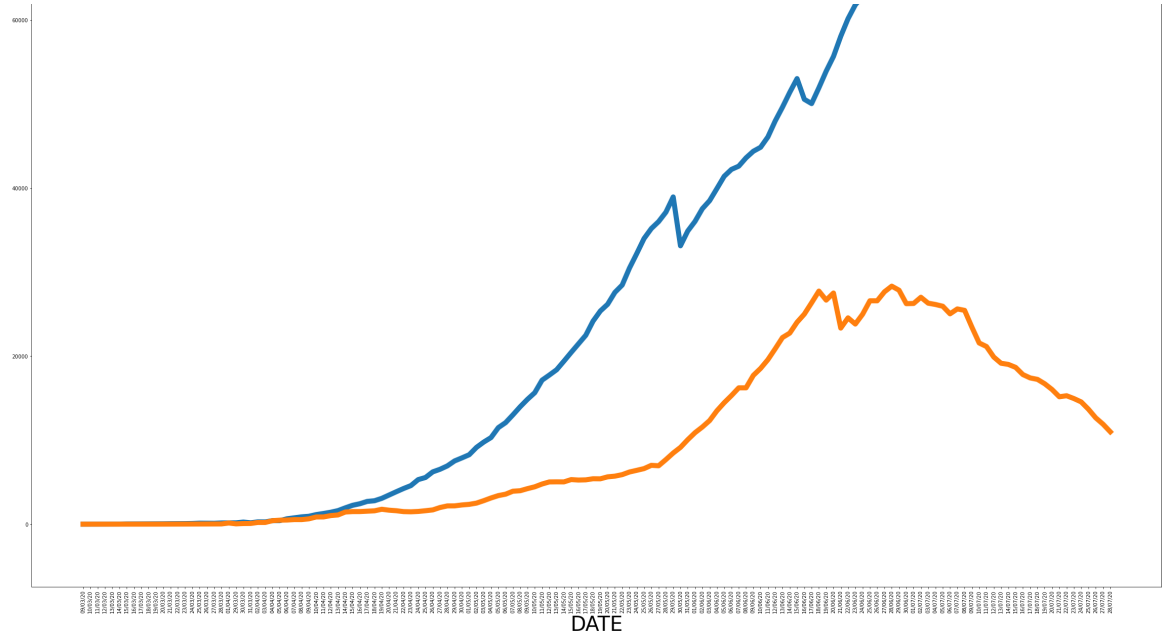
Out[28]:
```
<matplotlib.legend.Legend at 0x20722353940>
```



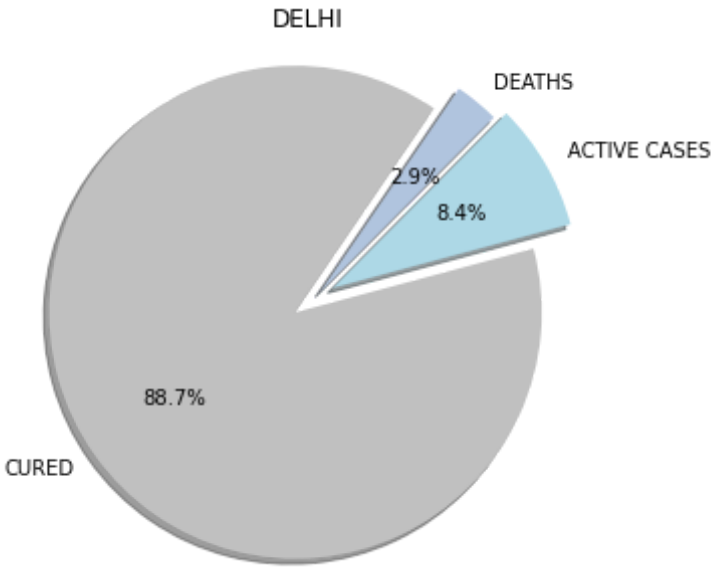
In [27]:
```
print("ACTIVE vs DATE")
x=df1_sv['Date'].tail(142)
y4=df1_sv['Active'].tail(142)
plt.xticks(rotation=90)
y5=df1_sv1['Active'].tail(142)
plt.xlabel('DATE', fontsize=40)
plt.ylabel('ACTIVE', fontsize=100)
plt.rc('legend', fontsize='xx-large')
plt.plot(x,y5,x,y4,linewidth=10)
plt.legend(['Maharastra', 'Delhi'])
```

```
ACTIVE vs DATE
```

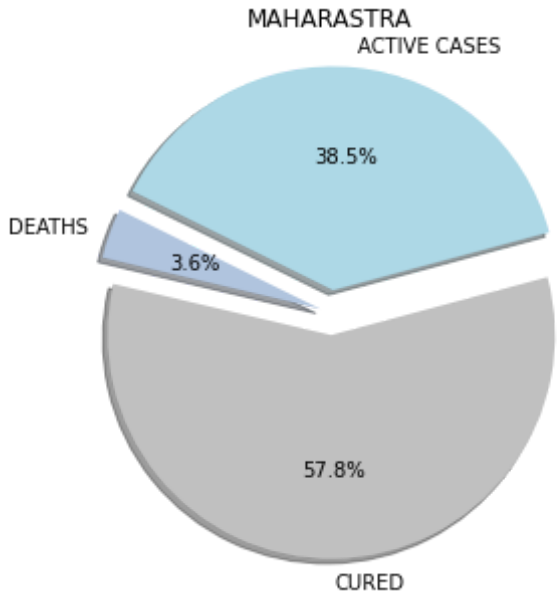
Out[27]:
```
<matplotlib.legend.Legend at 0x20721f27f10>
```



In [37]:
```
plt.rcParams["figure.figsize"] = (10,5)
p=df1_sv.iloc[-1]['Active']
q=df1_sv.iloc[-1]['Deaths']
r=df1_sv.iloc[-1]['Cured']
task=[p,q,r]
my_labels = 'ACTIVE CASES','DEATHS','CURED'
my_colors = ['lightblue','lightsteelblue','silver']
my_explode = (0.08, 0.03, 0.1)
plt.pie(task, labels=my_labels, autopct='%1.1f%%', startangle=15, shado
w = True, colors=my_colors, explode=my_explode)
plt.title('DELHI')
plt.axis('equal')
plt.show()
```



In [38]:
```
p=df1_sv1.iloc[-1]['Active']
q=df1_sv1.iloc[-1]['Deaths']
r=df1_sv1.iloc[-1]['Cured']
task=[p,q,r]
my_labels = 'ACTIVE CASES','DEATHS','CURED'
my_colors = ['lightblue','lightsteelblue','silver']
my_explode = (0.1, 0.05, 0.1)
plt.pie(task, labels=my_labels, autopct='%1.1f%%', startangle=15, shado
w = True, colors=my_colors, explode=my_explode)
plt.title('MAHARASTRA')
plt.axis('equal')
plt.show()
```



In [39]:
```
print("DELHI : ")
print("FROM THE ABOVE TWO PIE CHARTS WE CAN COMPARE AND CONCLUDE THAT 8
8.7% PEOPLE IN DELHI GOT RECOVERED FROM THE DISEASE OUT OF THE TOTAL NU
MBER OF CASES. WHEREAS THE ACTIVE CASES PERCENTAGE REMAINS 8.4%. 2.9% P
EOPLE OUT OF THE TOTAL NUMBER OF CASES LOST THEIR LIVES TO THIS PANDEMI
C IN DELHI")
print("\n")
print("MUMBAI : ")
print("IN MUMBAI WE CAN NOTICE THAT ONLY 57.8% PEOPLE GOT CURED OUT OF
 THE TOTAL NUMBER OF CASES. 38.5% PEOPLE STILL REMAIN AFFECTED TO THIS
 VIRUS. 3.6% OF PEOPLE LOST THEIR LIVES OUT OF THE TOTAL NUMBER OF CASE
S TO THIS DISEASE.")
```

```
DELHI :
FROM THE ABOVE TWO PIE CHARTS WE CAN COMPARE AND CONCLUDE THAT 88.7% PE
OPLE IN DELHI GOT RECOVERED FROM THE DISEASE OUT OF THE TOTAL NUMBER OF
CASES. WHEREAS THE ACTIVE CASES PERCENTAGE REMAINS 8.4%. 2.9% PEOPLE OU
T OF THE TOTAL NUMBER OF CASES LOST THEIR LIVES TO THIS PANDEMIC IN DEL
HI
```

MUMBAI :

IN MUMBAI WE CAN NOTICE THAT ONLY 57.8% PEOPLE GOT CURED OUT OF THE TOTAL NUMBER OF CASES. 38.5% PEOPLE STILL REMAIN AFFECTED TO THIS VIRUS. 3.6% OF PEOPLE LOST THEIR LIVES OUT OF THE TOTAL NUMBER OF CASES TO THIS DISEASE.

In [ ]: