In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib
from matplotlib.ticker import AutoMinorLocator,MultipleLocator,FuncFormatter,LinearLocator,NullLocator,FixedLocator,IndexL
```

利用pd.read_csv()读取上步bash脚本中生成的各个脚本

In [2]:

```
seq_AVL=pd.read_csv('../time/AVL_seq_time.csv')
seq_Splay=pd.read_csv('../time/Splayref_seq_time.csv')
seq_unbalanced=pd.read_csv('../time/unbalancedref_seq_time.csv')
anti_AVL=pd.read_csv('../time/AVL_anti_time.csv')
anti_Splay=pd.read_csv('../time/Splayref_anti_time.csv')
anti_unbalanced=pd.read_csv('../time/unbalancedref_anti_time.csv')
random_AVL=pd.read_csv('../time/AVL_random_time.csv')
random_Splay=pd.read_csv('../time/Splayref_random_time.csv')
random_unbalanced=pd.read_csv('../time/unbalancedref_random_time.csv')
```

重新组织dataframe结构,并且将dataframe的index从(0,9001)转成(1000,10001)

In [3]:

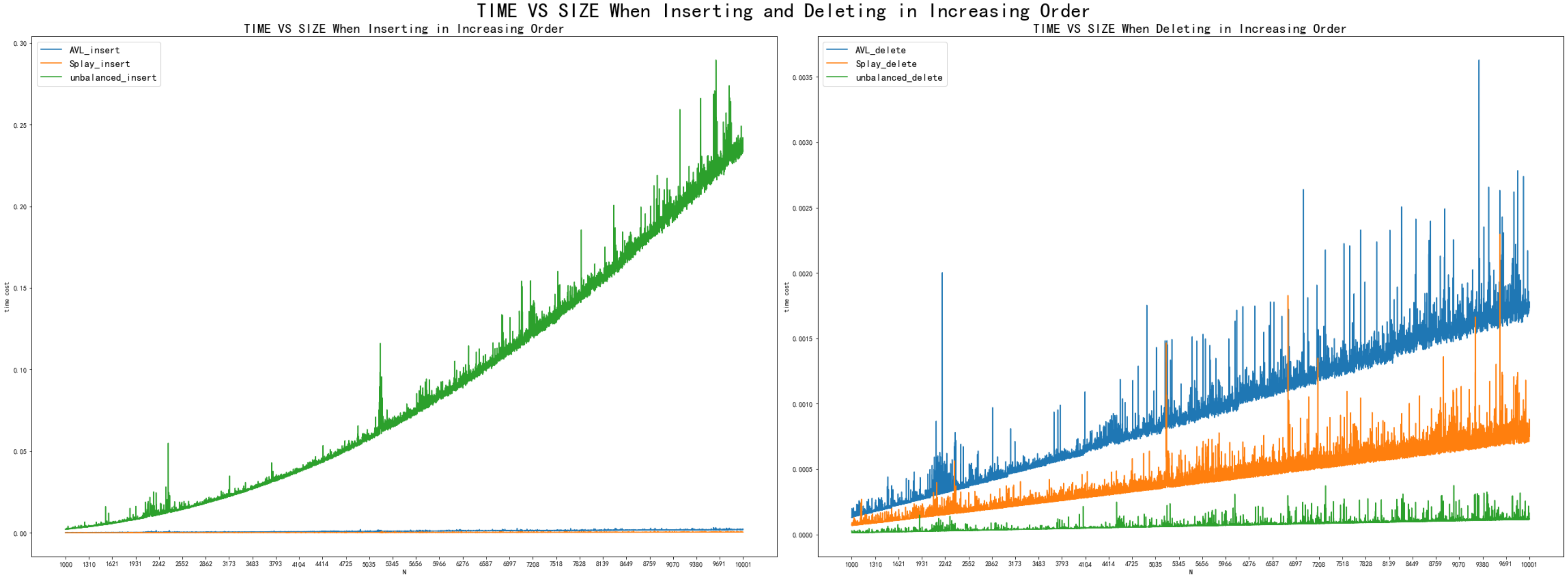
```
seq_insert=pd.DataFrame({'AVL_insert':seq_AVL['insert_time'],
                     'Splay_insert':seq_Splay['insert_time'],
                     'unbalanced_insert':seq_unbalanced['insert_time']})
seq_delete=pd.DataFrame({'AVL_delete':seq_AVL['delete_time'],
                        'Splay_delete':seq_Splay['delete_time'],
                        'unbalanced_delete':seq_unbalanced['delete_time']})
anti_insert=pd.DataFrame({'AVL_insert':anti_AVL['insert_time'],
                     'Splay_insert':anti_Splay['insert_time'],
                     'unbalanced_insert':anti_unbalanced['insert_time']})
anti_delete=pd.DataFrame({'AVL_delete':anti_AVL['delete_time'],
                        'Splay_delete':anti_Splay['delete_time'],
                        'unbalanced_delete':anti_unbalanced['delete_time']})
random_insert=pd.DataFrame({'AVL_insert':random_AVL['insert_time'],
                     'Splay_insert':random_Splay['insert_time'],
                     'unbalanced_insert':random_unbalanced['insert_time']})
random_delete=pd.DataFrame({'AVL_delete':random_AVL['delete_time'],
                        'Splay_delete':random_Splay['delete_time'],
                        'unbalanced_delete':random_unbalanced['delete_time']})
AVL=pd.DataFrame({'insert':seq_AVL['insert_time'],
                'random_insert':random_AVL['insert_time'],
                'seq_delete':seq_AVL['delete_time'],
                'anti_delete':anti_AVL['delete_time'],
                'random_delete':random_AVL['delete_time']
               })
```

```
Splay=pd.DataFrame({'insert':seq_Splay['insert_time'],
                    'random_insert':random_Splay['insert_time'],
                  'seq_delete':seq_Splay['delete_time'],
                  'anti_delete':anti_Splay['delete_time'],
                  'random_delete':random_Splay['delete_time']
                 })
unbalanced=pd.DataFrame({'insert':seq_unbalanced['insert_time'],
                         'random_insert':random_unbalanced['insert_time'],
                  'seq_delete':seq_unbalanced['delete_time'],
                  'anti_delete':anti_unbalanced['delete_time'],
                  'random_delete':random_unbalanced['delete_time']
                 })
seq_insert.index=range(1000,10001,1)
seq_delete.index=range(1000,10001,1)
anti_insert.index=range(1000,10001,1)
anti_delete.index=range(1000,10001,1)
random_insert.index=range(1000,10001,1)
random_delete.index=range(1000,10001,1)
AVL.index=range(1000,10001,1)
Splay.index=range(1000,10001,1)
unbalanced.index=range(1000,10001,1)
```

分别取出三种算法顺序插入与顺序删除的时间列表，以此为y值作图，并将图保存在指定路径下

In [4]:

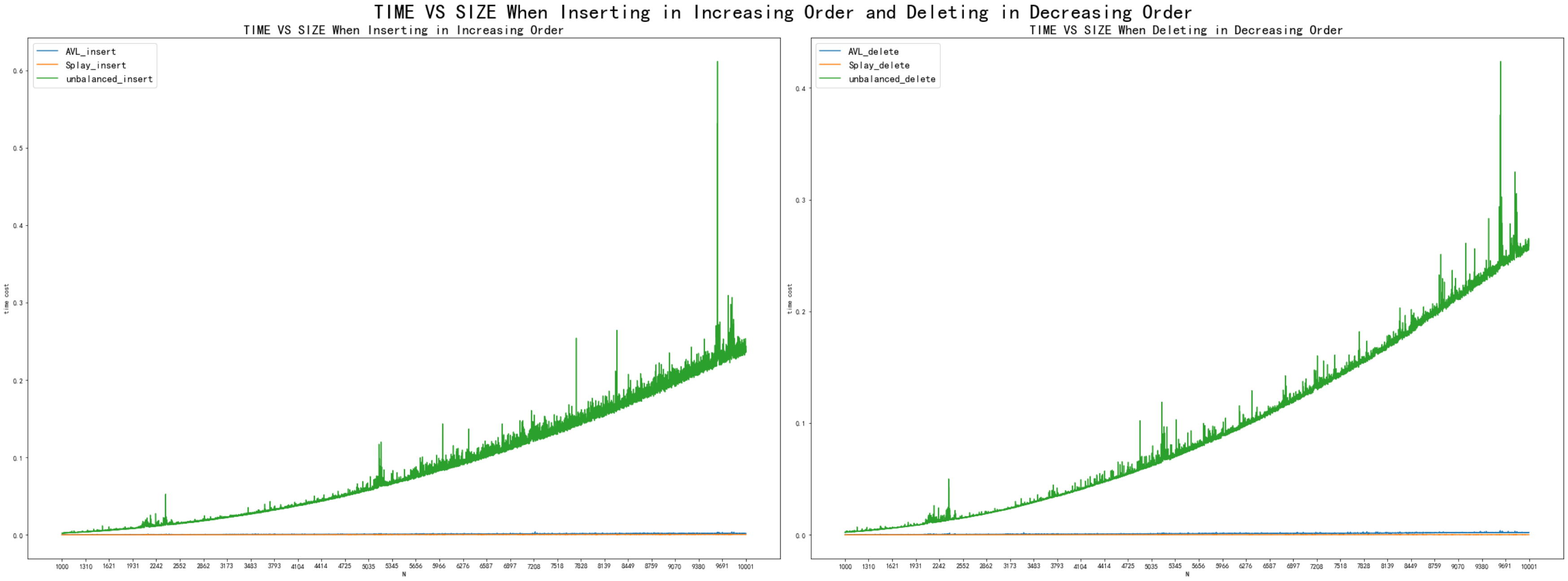
```
plt.rcParams['figure.figsize'] = (32.0, 12.0) # 单位是inches
plt.suptitle('TIME VS SIZE When Inserting and Deleting in Increasing Order',fontsize=30)
xt=np.linspace(1000,10001,30)
ax1=plt.subplot(1,2,1)

# data_insert.plot()
#第一幅子图，设置字体、图例、标题、坐标轴刻度
plt.plot(seq_insert['AVL_insert'],label='AVL_insert')
plt.plot(seq_insert['Splay_insert'],label='Splay_insert')
plt.plot(seq_insert['unbalanced_insert'],label='unbalanced_insert')

plt.title("TIME VS SIZE When Inserting in Increasing Order",fontsize=20)
plt.xlabel("N")
plt.ylabel("time cost")
plt.legend(loc="best",fontsize=15)
plt.xticks(xt)
ax2=plt.subplot(1,2,2)
# data_delete.plot()
#第二幅子图，设置字体、图例、标题、坐标轴刻度
plt.plot(seq_delete['AVL_delete'],label='AVL_delete')
plt.plot(seq_delete['Splay_delete'],label='Splay_delete')
plt.plot(seq_delete['unbalanced_delete'],label='unbalanced_delete')
plt.title("TIME VS SIZE When Deleting in Increasing Order",fontsize=20)
plt.xlabel("N")
plt.ylabel("time cost")
plt.legend(loc="best",fontsize=15)
plt.xticks(xt)
plt.tight_layout()
#将图存成png文件

plt.savefig("../pictures/insert&seq_delete.png")
plt.show()
```



分别取出三种算法顺序插入与逆序删除的时间列表，以此为y值作图，并将图保存在指定路径下

In [5]:

```
plt.rcParams['figure.figsize'] = (32.0, 12.0) # 单位是inches
plt.suptitle('TIME VS SIZE When Inserting in Increasing Order and Deleting in Decreasing Order',fontsize=30)
xt=np.linspace(1000,10001,30)
ax1=plt.subplot(1,2,1)

# data_insert.plot()
plt.plot(anti_insert['AVL_insert'],label='AVL_insert')
plt.plot(anti_insert['Splay_insert'],label='Splay_insert')
plt.plot(anti_insert['unbalanced_insert'],label='unbalanced_insert')
plt.title("TIME VS SIZE When Inserting in Increasing Order",fontsize=20)
plt.xlabel("N")
plt. ylabel(″time cost″)
plt. legend(loc=″best″, fontsize=15)
plt. xticks(xt)
ax2=plt. subplot(1, 2, 2)
# data_delete. plot()
plt. plot(anti_delete[’AVL_delete’], label=’AVL_delete’)
plt. plot(anti_delete[’Splay_delete’], label=’Splay_delete’)
plt. plot(anti_delete[’unbalanced_delete’], label=’unbalanced_delete’)
plt. title(″TIME VS SIZE When Deleting in Decreasing Order″, fontsize=20)
plt. xlabel(″N″)
plt. ylabel(″time cost″)
plt. legend(loc=″best″, fontsize=15)
plt. xticks(xt)
plt. tight_layout()


plt. savefig(″../pictures/insert&anti_delete.png″)
plt. show()
```



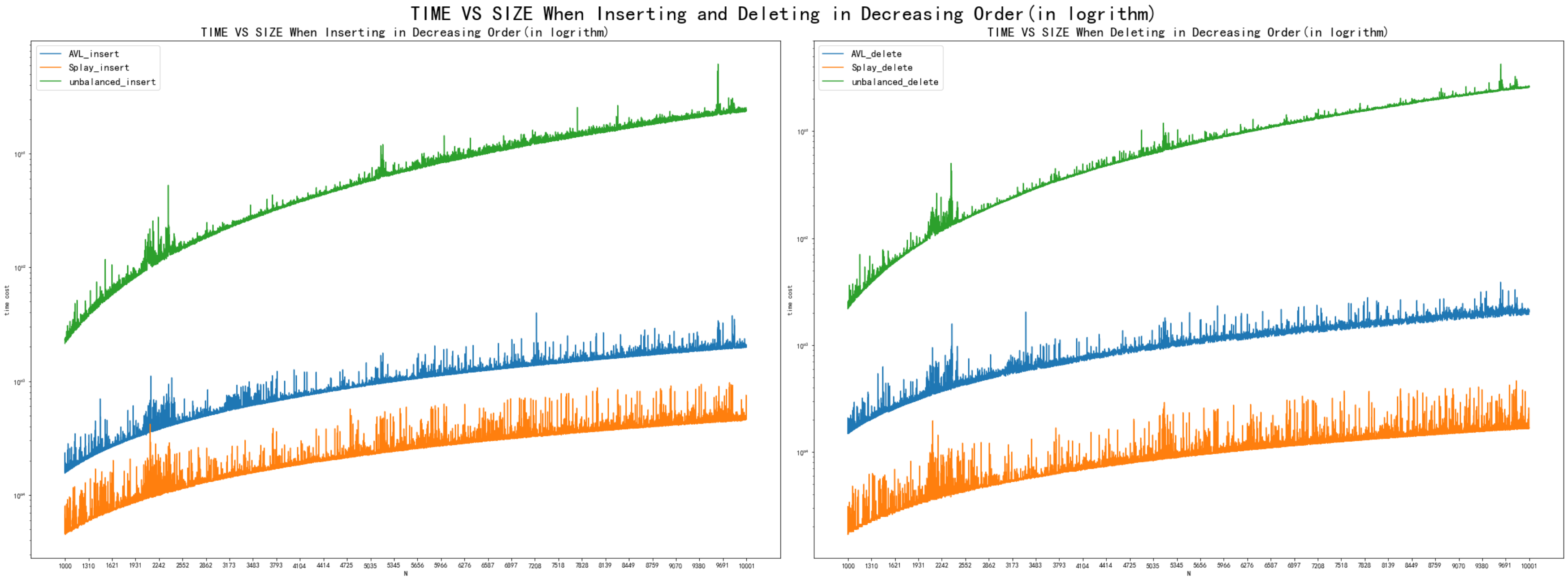
由上述图中可以看出，这种情况下unbalance tree与其余两种算法差了几个数量级，因此对时间做对数处理 分别取出三种算法顺序插入与逆序删除的时间列表，取对数后以此为y值作图，并将图保存在指定路径下

In [6]:

```
plt. rcParams[’figure. figsize’] = (32.0, 12.0) # 单位是inches
plt. suptitle(’TIME VS SIZE When Inserting and Deleting in Decreasing Order(in logrithm)’, fontsize=30)
```

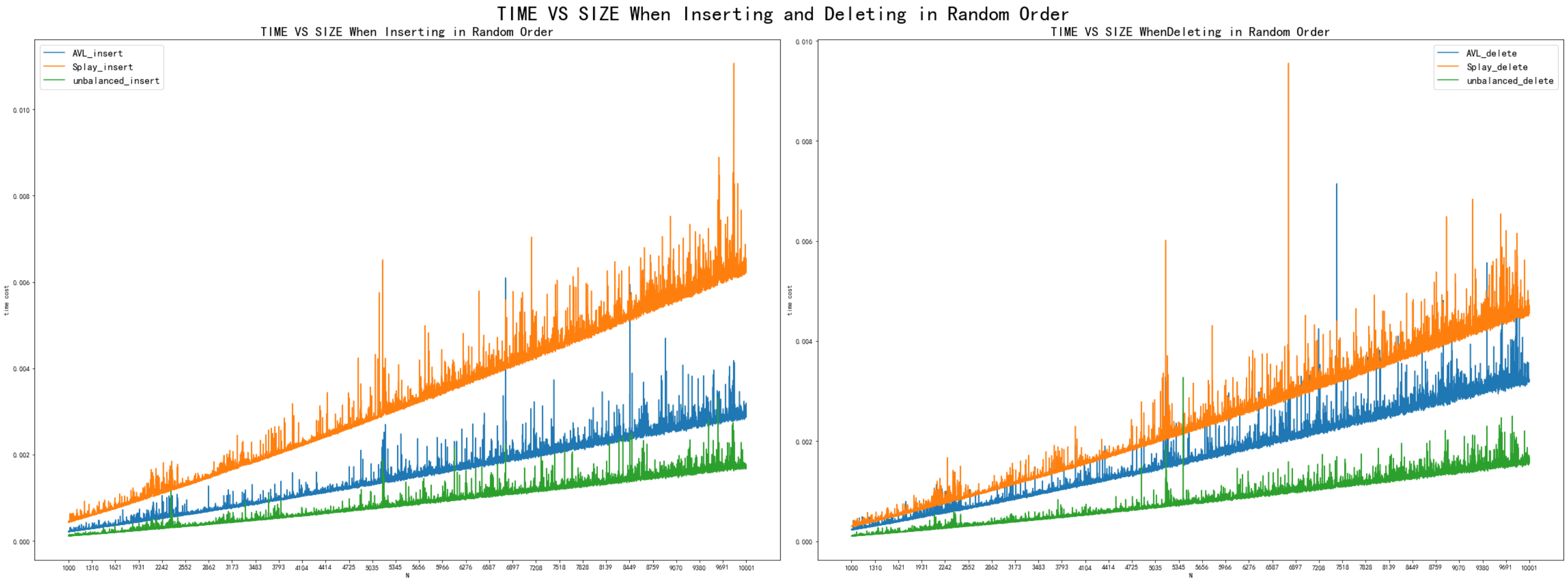
```
xt=np.linspace(1000,10001,30)
ax1=plt.subplot(1,2,1)

# data_insert.plot()
plt.plot(anti_insert['AVL_insert'],label='AVL_insert')
plt.plot(anti_insert['Splay_insert'],label='Splay_insert')
plt.plot(anti_insert['unbalanced_insert'],label='unbalanced_insert')
plt.title("TIME VS SIZE When Inserting in Decreasing Order(in logrithm)",fontsize=20)
plt.xlabel("N")
plt.ylabel("time cost")
plt.legend(loc="best",fontsize=15)
plt.xticks(xt)
plt.yscale('log')
ax2=plt.subplot(1,2,2)
# data_delete.plot()
plt.plot(anti_delete['AVL_delete'],label='AVL_delete')
plt.plot(anti_delete['Splay_delete'],label='Splay_delete')
plt.plot(anti_delete['unbalanced_delete'],label='unbalanced_delete')
plt.title("TIME VS SIZE When Deleting in Decreasing Order(in logrithm)",fontsize=20)
plt.xlabel("N")
plt.ylabel("time cost")
plt.legend(loc="best",fontsize=15)
plt.xticks(xt)
plt.yscale('log')
plt.tight_layout()


plt.savefig("../pictures/insert&anti_delete_log.png")
plt.show()
```

```
Font 'default' does not have a glyph for '-' [U+2212], substituting with a dummy symbol.
Font 'default' does not have a glyph for '-' [U+2212], substituting with a dummy symbol.
Font 'default' does not have a glyph for '-' [U+2212], substituting with a dummy symbol.
Font 'default' does not have a glyph for '-' [U+2212], substituting with a dummy symbol.
Font 'default' does not have a glyph for '-' [U+2212], substituting with a dummy symbol.
Font 'default' does not have a glyph for '-' [U+2212], substituting with a dummy symbol.
Font 'default' does not have a glyph for '-' [U+2212], substituting with a dummy symbol.
Font 'default' does not have a glyph for '-' [U+2212], substituting with a dummy symbol.
Font 'default' does not have a glyph for '-' [U+2212], substituting with a dummy symbol.
Font 'default' does not have a glyph for '-' [U+2212], substituting with a dummy symbol.
Font 'default' does not have a glyph for '-' [U+2212], substituting with a dummy symbol.
Font 'default' does not have a glyph for '-' [U+2212], substituting with a dummy symbol.
Font 'default' does not have a glyph for '-' [U+2212], substituting with a dummy symbol.
Font 'default' does not have a glyph for '-' [U+2212], substituting with a dummy symbol.
Font 'default' does not have a glyph for '-' [U+2212], substituting with a dummy symbol.
Font 'default' does not have a glyph for '-' [U+2212], substituting with a dummy symbol.
```



分别取出三种算法随机插入与随机删除的时间列表，取对数后以此为y值作图，并将图保存在指定路径下

In [7]:
```
plt.rcParams['figure.figsize'] = (32.0, 12.0) # 单位是inches
plt.suptitle('TIME VS SIZE When Inserting and Deleting in Random Order',fontsize=30)
xt=np.linspace(1000,10001,30)
ax1=plt.subplot(1,2,1)

# data_insert.plot()
plt.plot(random_insert['AVL_insert'],label='AVL_insert')
plt.plot(random_insert['Splay_insert'],label='Splay_insert')
plt.plot(random_insert['unbalanced_insert'],label='unbalanced_insert')
plt.title("TIME VS SIZE When Inserting in Random Order",fontsize=20)
plt.xlabel("N")
plt.ylabel("time cost")
plt.legend(loc="best",fontsize=15)
plt.xticks(xt)
ax2=plt.subplot(1,2,2)
# data_delete.plot()
plt.plot(random_delete['AVL_delete'],label='AVL_delete')
plt.plot(random_delete['Splay_delete'],label='Splay_delete')
plt.plot(random_delete['unbalanced_delete'],label='unbalanced_delete')

plt.title("TIME VS SIZE WhenDeleting in Random Order",fontsize=20)
plt.xlabel("N")
plt. ylabel("time cost")
plt. legend(loc="best", fontsize=15)
plt. xticks(xt)
plt. tight_layout()


plt. savefig("../pictures/insert&random_delete.png")
plt. show()
```



TIME VS SIZE When Inserting and Deleting in Random Order

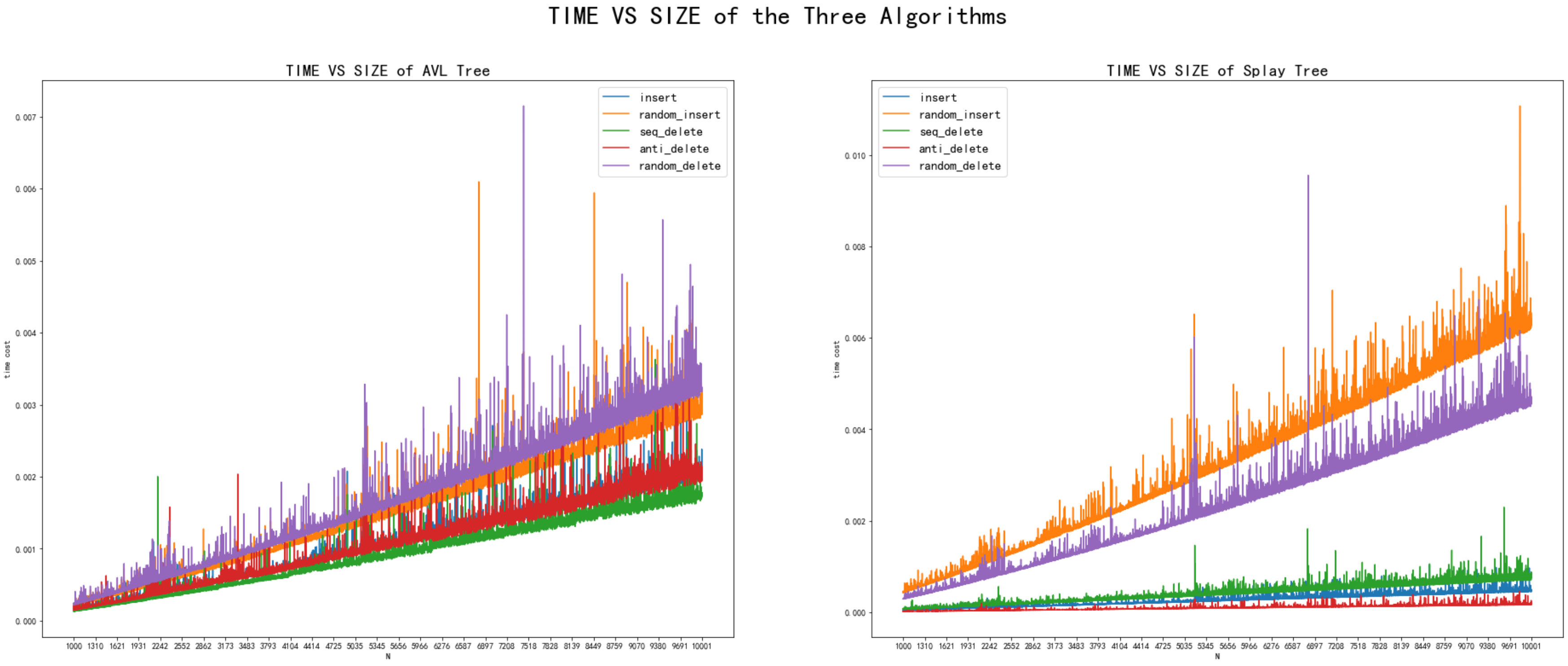
TIME VS SIZE When Inserting in Random Order

TIME VS SIZE WhenDeleting in Random Order

In [8]:

```
plt. suptitle('TIME VS SIZE of the Three Algorithms', fontsize=30)
xt=np. linspace(1000, 10001, 30)
ax1=plt. subplot(1, 2, 1)

# data_insert. plot()
plt. plot(AVL['insert'], label='insert')
plt. plot(AVL['random_insert'], label='random_insert')
plt. plot(AVL['seq_delete'], label='seq_delete')
plt. plot(AVL['anti_delete'], label='anti_delete')
plt. plot(AVL['random_delete'], label='random_delete')
plt. title("TIME VS SIZE of AVL Tree", fontsize=20)
plt. xlabel("N")
plt. ylabel("time cost")
plt. legend(loc="best", fontsize=15)
plt. xticks(xt)
ax2=plt. subplot(1, 2, 2)
```

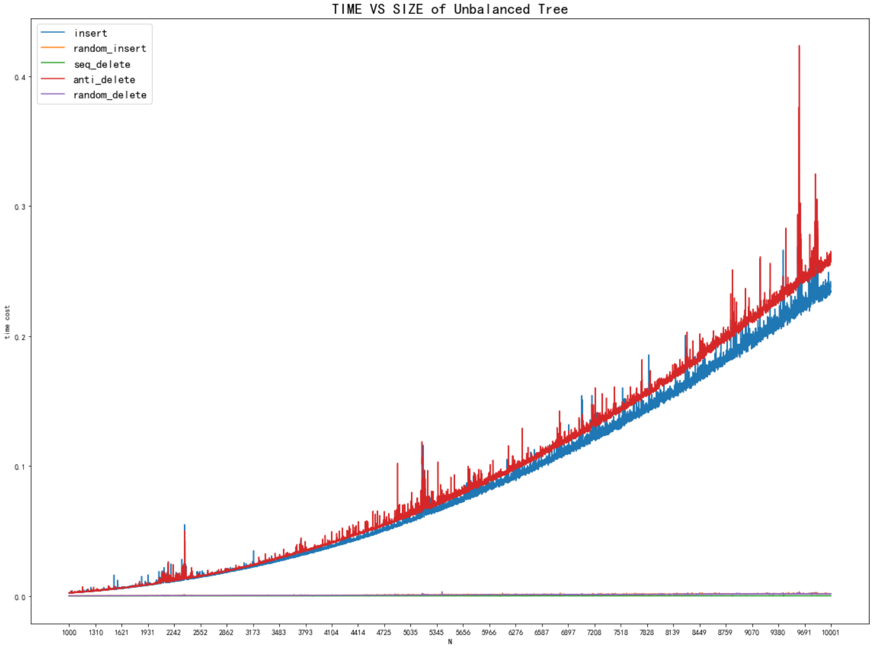
```
# data_delete.plot()
plt.plot(Splay[’insert’],label=’insert’)
plt.plot(Splay[’random_insert’],label=’random_insert’)
plt.plot(Splay[’seq_delete’],label=’seq_delete’)
plt.plot(Splay[’anti_delete’],label=’anti_delete’)
plt.plot(Splay[’random_delete’],label=’random_delete’)
plt.title(”TIME VS SIZE of Splay Tree”,fontsize=20)
plt.xlabel(”N”)
plt.ylabel(”time cost”)
plt.legend(loc=”best”,fontsize=15)
plt.xticks(xt)


plt.savefig(”../pictures/AVL&Splay.png”)
plt.show()
```

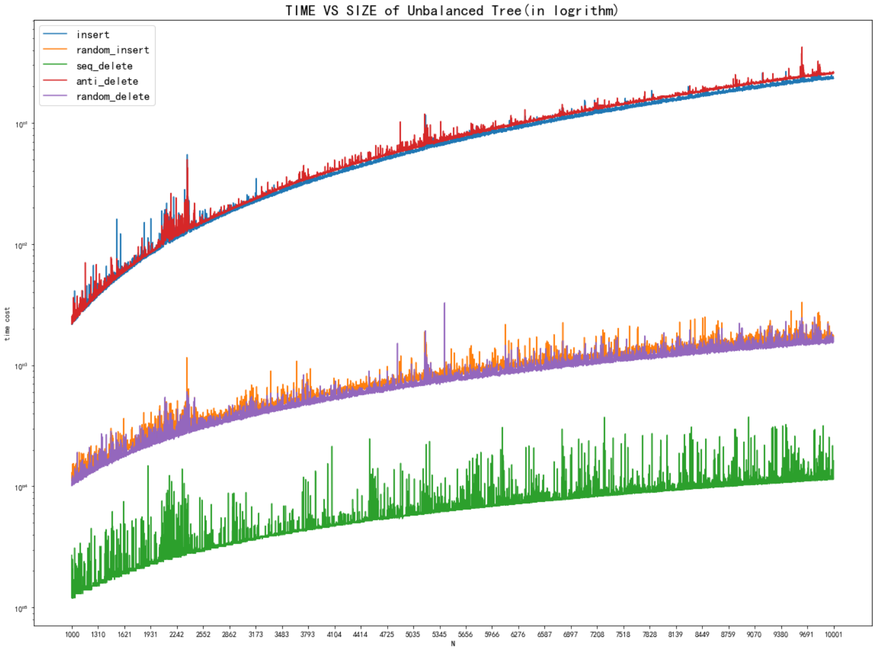


In [9]:

```
ax2=plt.subplot(1,2,1)
# data_delete.plot()
plt.plot(unbalanced[’insert’],label=’insert’)
plt.plot(unbalanced[’random_insert’],label=’random_insert’)
plt.plot(unbalanced[’seq_delete’],label=’seq_delete’)
plt.plot(unbalanced[’anti_delete’],label=’anti_delete’)
plt.plot(unbalanced['random_delete'],label='random_delete')
plt.title("TIME VS SIZE of Unbalanced Tree",fontsize=20)
plt.xlabel("N")
plt.ylabel("time cost")
plt.legend(loc="best",fontsize=15)
plt.xticks(xt)

ax2=plt.subplot(1,2,2)
# data_delete.plot()
plt.plot(unbalanced['insert'],label='insert')
plt.plot(unbalanced['random_insert'],label='random_insert')
plt.plot(unbalanced['seq_delete'],label='seq_delete')
plt.plot(unbalanced['anti_delete'],label='anti_delete')
plt.plot(unbalanced['random_delete'],label='random_delete')
plt.title("TIME VS SIZE of Unbalanced Tree(in logrithm)",fontsize=20)
plt.xlabel("N")
plt.ylabel("time cost")
plt.legend(loc="best",fontsize=15)
plt.xticks(xt)
plt.yscale('log')

plt.tight_layout()


plt.savefig("../pictures/unbalanced_with_log.png")
plt.show()
```

```
Font 'default' does not have a glyph for '−' [U+2212], substituting with a dummy symbol.
Font 'default' does not have a glyph for '−' [U+2212], substituting with a dummy symbol.
Font 'default' does not have a glyph for '−' [U+2212], substituting with a dummy symbol.
Font 'default' does not have a glyph for '−' [U+2212], substituting with a dummy symbol.
Font 'default' does not have a glyph for '−' [U+2212], substituting with a dummy symbol.
Font 'default' does not have a glyph for '−' [U+2212], substituting with a dummy symbol.
Font 'default' does not have a glyph for '−' [U+2212], substituting with a dummy symbol.
Font 'default' does not have a glyph for '−' [U+2212], substituting with a dummy symbol.
Font 'default' does not have a glyph for '−' [U+2212], substituting with a dummy symbol.
Font 'default' does not have a glyph for '−' [U+2212], substituting with a dummy symbol.
Font 'default' does not have a glyph for '−' [U+2212], substituting with a dummy symbol.
Font 'default' does not have a glyph for '−' [U+2212], substituting with a dummy symbol.
Font 'default' does not have a glyph for '−' [U+2212], substituting with a dummy symbol.
Font 'default' does not have a glyph for '−' [U+2212], substituting with a dummy symbol.
Font 'default' does not have a glyph for '−' [U+2212], substituting with a dummy symbol.
Font 'default' does not have a glyph for '−' [U+2212], substituting with a dummy symbol.
Font 'default' does not have a glyph for '−' [U+2212], substituting with a dummy symbol.
Font 'default' does not have a glyph for '−' [U+2212], substituting with a dummy symbol.
Font 'default' does not have a glyph for '−' [U+2212], substituting with a dummy symbol.
Font 'default' does not have a glyph for '−' [U+2212], substituting with a dummy symbol.
```


TIME VS SIZE of Unbalanced Tree
insert
random_insert
seq_delete
anti_delete
random_delete
time cost
0.4
0.3
0.2
0.1
0.0
1000
1310
1621
1931
2242
2552
2862
3173
3483
3793
4104
4414
4725
5035
5345
5656
5966
6276
6587
6897
7208
7518
7828
8139
8449
8759
9070
9380
9691
10001
N


TIME VS SIZE of Unbalanced Tree(in logrithm)
insert
random_insert
seq_delete
anti_delete
random_delete
time cost
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
1000
1310
1621
1931
2242
2552
2862
3173
3483
3793
4104
4414
4725
5035
5345
5656
5966
6276
6587
6897
7208
7518
7828
8139
8449
8759
9070
9380
9691
10001
N