IP Bisect Question

Imports

In [1]:
```
import pandas as pd
from bisect import bisect
```

Consider the following theoretical DataFrame that stores a very simplified version of our "ips" DataFrame:

In [2]:
```
ip_df = pd.DataFrame({"low":[float(x) for x in range(5)],
                      "high":[float(f"{x}.9") for x in range(5)],
                      "region": ["UK", "Norway", "USA", "Ireland", "France"]})
ip_df
```

Out[2]:

	low	high	region
0	0.0	0.9	UK
1	1.0	1.9	Norway
2	2.0	2.9	USA
3	3.0	3.9	Ireland
4	4.0	4.9	France

Also, suppose that our IP address is `2.3` which clearly falls under being a USA IP address. When we are doing our lookup, we are only looking at the values in `ip_df["low"]`.

In [3]:
```
our_ip = 2.3
ip_df["low"]
```

Out[3]:
```
0    0.0
1    1.0
2    2.0
3    3.0
4    4.0
Name: low, dtype: float64
```

Now, when we call `bisect(ip_df["low"], our_ip)`, we get the index value 3. This is because if we place `2.3` at the index 3 of `ip_df["low"]`, it remains sorted.

In [4]:
```
idx = bisect(ip_df["low"], our_ip)
idx
```

Out[4]:
```
3
```

Note how the list remains sorted.

In [5]:
```
example_list = list(ip_df["low"])
example_list.insert(idx, our_ip)
```

```
example_list
```

Out[5]:
```
[0.0, 1.0, 2.0, 2.3, 3.0, 4.0]
```

Remember that we didn't actually insert our value (and we will never actually insert it, we are just using it for look-ups).

So when we do `ip_df.iloc[idx, "region"]` we are going to get the value in row 3 (since $2.0 \leq \text{our\_ip} = 2.3 < 3.0$). This is why we get Ireland, because we are getting the row at which all of the ***low*** values are less than or equal to our ip.

In [6]:
```
ip_df.iloc[idx]["region"]
```

Out[6]:
```
'Ireland'
```

Now, since we are always going to get the region that is one index "higher" than our actual index, we simply need to subtract 1 from our index.

In [7]:
```
ip_df.iloc[idx - 1]["region"]
```

Out[7]:
```
'USA'
```

TL;DR

Using bisect gives us the index where we would insert our value to keep sorted order. We never actually insert our value, so we get the row about the one we want. To fix this, we just need to subtract 1 from the index we get from bisect.

In [8]:
```
ip_df = pd.DataFrame({"low":[float(x) for x in range(5)],
                      "high":[float(f"{x}.9") for x in range(5)],
                      "region": ["UK", "Norway", "USA", "Ireland", "France"]})
our_ip = 2.3
idx = bisect(ip_df["low"], our_ip)
print(f"ip_df.iloc[idx]['region']:     {ip_df.iloc[idx]['region']}")
print(f"ip_df.iloc[idx - 1]['region']: {ip_df.iloc[idx - 1]['region']}")
```

```
ip_df.iloc[idx]['region']:     Ireland
ip_df.iloc[idx - 1]['region']: USA
```