

Upper Confidence Bound (USB)

```
In [1]: # Importing the Libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

In [2]: # Importing the dataset
dataset = pd.read_csv('ads CTR optimisation.csv')
```

```
In [3]: dataset.head()
```

Out[3]:

	Ad 1	Ad 2	Ad 3	Ad 4	Ad 5	Ad 6	Ad 7	Ad 8	Ad 9	Ad 10
0	1	0	0	0	1	0	0	0	1	0
1	0	0	0	0	0	0	0	0	1	0
2	0	0	0	0	0	0	0	0	0	0
3	0	1	0	0	0	0	0	1	0	0
4	0	0	0	0	0	0	0	0	0	0

```
In [4]: dataset.tail()
```

Out[4]:

	Ad 1	Ad 2	Ad 3	Ad 4	Ad 5	Ad 6	Ad 7	Ad 8	Ad 9	Ad 10
9995	0	0	1	0	0	0	0	1	0	0
9996	0	0	0	0	0	0	0	0	0	0
9997	0	0	0	0	0	0	0	0	0	0
9998	1	0	0	0	0	0	0	1	0	0
9999	0	1	0	0	0	0	0	0	0	0

```
In [5]: dataset.describe()
```

Out[5]:

	Ad 1	Ad 2	Ad 3	Ad 4	Ad 5	Ad 6	Ad 7	Ad 8	Ad 9	Ad 10
count	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000
mean	0.170300	0.129500	0.072800	0.119600	0.269500	0.012600	0.111200	0.209100	0.095200	0.04890
std	0.375915	0.335769	0.259821	0.324509	0.443722	0.111546	0.314395	0.406686	0.293506	0.21567
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00000
25%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00000
50%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00000
75%	0.000000	0.000000	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.00000
max	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.00000

```
In [6]: #Checking Missing Values
dataset.isnull().sum()
```

Out[6]:

```
Ad 1      0
Ad 2      0
Ad 3      0
Ad 4      0
Ad 5      0
Ad 6      0
Ad 7      0
Ad 8      0
Ad 9      0
Ad 10     0
dtype: int64
```

Implementing USB

```
In [7]: import math
N = 5000
d = 10
ads_selected = []
numbers_of_selections = [0] * d
sum_of_rewards = [0] * d
total_reward = 0
for n in range(0,N):
    ad = 0
    max_upper_bound = 0
    for i in range(0,d):
        if(numbers_of_selections[i] > 0):
            average_reward = sum_of_rewards[i] / numbers_of_selections[i]
            delta_i = math.sqrt(3/2 * math.log(n + 1) / numbers_of_selections[i])
            upper_bound = average_reward + delta_i

        else:
            upper_bound = 1e400

        if upper_bound > max_upper_bound:
            max_upper_bound = upper_bound
            ad = i

    ads_selected.append(ad)
    numbers_of_selections[ad] = numbers_of_selections[ad] + 1
    reward = dataset.values[n,ad]
    sum_of_rewards[ad] = sum_of_rewards[ad] + reward
    total_reward = total_reward + reward
```

Visualising the results - Histogram

```
In [8]: plt.hist(ads_selected , edgecolor = 'black')
plt.title('Histogram of ads selections')
plt.xlabel('Ads')
plt.ylabel('Number of times each ad was selected')
plt.show()
```

