In [1]:

```
import numpy as np
import matplotlib.pyplot as plt
import numpy.random as rnd
from scipy.stats import beta
```

In [2]:

```
T = 1000
R = 30
N = 6
d = 0

rewards_history = []
allocation_history = []
sum_reward_list = []

mu = np.array([0.3, 0.5, 0.4, 0.45, 0.3, 0.35])

for i in range(R):
    arms_allocation = [0] * N
    arms_reward = [0] * N
    total_reward = 0

    for n in range(0, T):
        max_upper_bound = 0
        upper_bounds = [0] * N
        lower_bounds = [0] * N

        for arm in range(0, N):
            if arms_allocation[arm] > 0:
                average_reward = arms_reward[arm] / arms_allocation[arm]
                delta = np.sqrt(2 * np.log1p(d) / arms_allocation[arm])
                upper_bound = average_reward + delta                                  ✓
                lower_bound = average_reward - delta
            else:
                upper_bound = 1e100
                lower_bound = 1e100

            if upper_bound > max_upper_bound:
                max_upper_bound = upper_bound
                chosen_arm = arm

        upper_bounds[arm] = upper_bound
        lower_bounds[arm] = lower_bound

        arms_allocation[chosen_arm] += 1
        action = arms_allocation[chosen_arm]
        allocation_history.append(arms_allocation[:])                               ✓
        reward = rnd.binomial(1, mu[chosen_arm]) # when n = 1, it works like bernouli
        arms_reward[chosen_arm] += reward
        rewards_history.append(arms_reward[:])
        d += 1

    sum_reward_list.append(np.sum(arms_reward))
```

In [3]:

```
for i in range(R):
    print("Time: ", i+1 , " Reward: ", sum_reward_list[i])
```

```
Time:  1  Reward:  400
Time:  2  Reward:  421
Time:  3  Reward:  416
Time:  4  Reward:  422
Time:  5  Reward:  420
Time:  6  Reward:  425
Time:  7  Reward:  436
Time:  8  Reward:  420
Time:  9  Reward:  401
Time:  10  Reward:  415
Time:  11  Reward:  422
Time:  12  Reward:  427
Time:  13  Reward:  422
Time:  14  Reward:  426
Time:  15  Reward:  416
Time:  16  Reward:  415
Time:  17  Reward:  438
Time:  18  Reward:  438
Time:  19  Reward:  419
Time:  20  Reward:  441
Time:  21  Reward:  397
Time:  22  Reward:  423
Time:  23  Reward:  437
Time:  24  Reward:  404
Time:  25  Reward:  415
Time:  26  Reward:  404
Time:  27  Reward:  391
Time:  28  Reward:  417
Time:  29  Reward:  422
Time:  30  Reward:  428
```

In [4]:

```
def sum_of_list(l):
    total = 0
    for val in l:
        total = total + val

    return total
```

In [5]:

```
Sum = sum_of_list(sum_reward_list)
Average = Sum/R
print("Average reward: ", Average)
```

```
Average reward:  419.26666666666665
```

In [ ]:

```
```

emp. Regret?