UCB1

In [ ]:
```
import pandas as pd,\
  utils.server_pull as usp,\
  numpy as np
```

Intialise desired pulls, API secret

In [ ]:
```
desired_pulls = 2000
team = ''
group_key = ''
```

Create function for getting top linUCB index of all arms

In [ ]:
```
def get_max_ucb_of_arms(input_df, input_global_round):
# Get the mean reward of each arm, and count of runs so far
# TODO: currently not safe for historical runs, would need indexing on inputdf
  df_mean_reward = input_df.pivot_table(
    index='arm',
    values=['arm_reward','arm_pull'],
    aggfunc={'arm_reward': np.mean, 'arm_pull': np.max }
  ).reset_index()
  df_mean_reward['ucb_index'] = df_mean_reward.apply(
    lambda x: x['arm_reward'] + np.sqrt(
      (2 * np.log( input_global_round ) ) / (x['arm_pull'] + 1)
    ),
    axis=1
  )
  return df_mean_reward['ucb_index'].idxmax()
# TODO: there is a non trivial circumstance where this can return multiple rows, rather than a scalar;
#   could be improved with getting random as typebreaker
```

Get historical data

In [ ]:

```
df_historical_pulls = pd.read_csv(
  './data/pulls.csv',
  header='infer',
  index_col=False
)
```

Get current arm pulls so far

In [ ]:

```
existing_arm_pulls = df_historical_pulls['global_pull'].max()
```

safety check if in exploration phase

In [ ]:

```
if existing_arm_pulls >= 23:
  # pre compute UCB for best arm
  target_arm = get_max_ucb_of_arms( df_historical_pulls, existing_arm_pulls )
  for i in range(existing_arm_pulls+1,existing_arm_pulls+desired_pulls+1):
    # pull arm, get output
    arm_output = usp.pull(team,group_key,target_arm)
    arm_pull_count = df_historical_pulls[df_historical_pulls['arm'] == target_arm  ]['arm_pull'].count()
    # append output
    df_historical_pulls = df_historical_pulls.append(
      {'arm_pull': arm_pull_count, 'arm':target_arm,'global_pull':i ,'arm_reward':arm_output['Reward'] },
      ignore_index=True
    )
    # get next arm
    target_arm = get_max_ucb_of_arms( df_historical_pulls,  i )
else:
  raise
```

write out results

```
In [ ]: df_historical_pulls.to_csv(
          './data/pulls_output.csv',
          index=False
        )
```

TODO: get arm pulls for exploration phase