- NAME:Rishabh Patil
- SAP: 60009200056
- BATCH: K2

Upper Confidence Bound

```
e=0.6
R=[0.5,0.25,0.4]
alpha=0.3
bounds=[1,1,1]
n=10
c=1
```

```
import numpy as np

def bandit(R,alpha,n):

  Q=[]
  Q.append([0])
  Q.append([0])
  Q.append([0])
  total=[]
  print(Q)
  import random

  arm1,arm2,arm3=0,0,0
  for i in range(1,n+1):
    new_Qka=max(Q[0])+alpha*(R[0]-Q[0][-1])
    new_Qkb=max(Q[1])+alpha*(R[1]-Q[1][-1])
    new_Qkc=max(Q[2])+alpha*(R[2]-Q[2][-1])

    if(new_Qka==R[0]):

      print("Arm 1 converged at: ",i)
    if(new_Qkb==R[1]):

      print("Arm 2 converged at: ",i)

    if(new_Qkc==R[2]):

      print("Arm 3 converged at: ",i)


    bounda=new_Qka+c*pow(np.log(i)/arm1,0.5)
    boundb=new_Qkb+c*pow(np.log(i)/arm2,0.5)
    boundc=new_Qkc+c*pow(np.log(i)/arm3,0.5)
    # print(bounda,boundb,boundc)
```

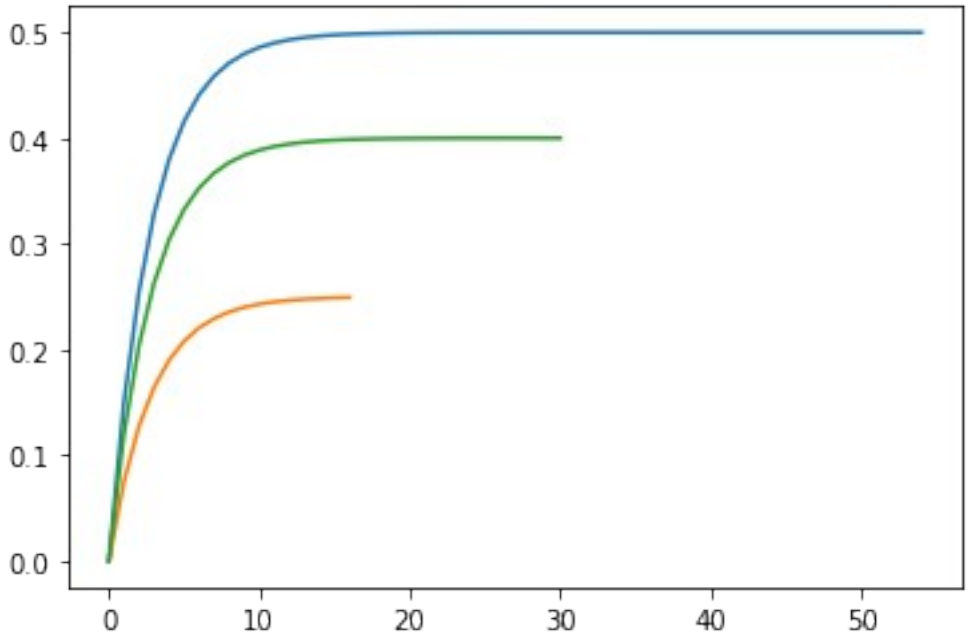
```
    maxi=np.argmax([bounda,boundb,boundc])
    # print(maxi)
    if(maxi==0):
      bounds[0]=bounda
      Q[0].append(new_Qka)
      arm1+=1
    elif(maxi==1):
      bounds[1]=boundb
      Q[1].append(new_Qkb)
      arm2+=1
    else:
      bounds[2]=boundc
      Q[2].append(new_Qkc)
      arm3+=1

    total.append(arm1*R[0]+arm2*R[1]+arm3*R[2])

    # print(Q)
print("Number of times arms are selected: ")
print(arm1,arm2,arm3)
print("The bound of a is: ",bounds[0])
print("The bound of b is: ",bounds[1])
print("The bound of c is: ",bounds[2])
print("Rewards for a are: ",arm1*R[0])
print("Rewards for b are: ",arm2*R[1])
print("Rewards for c are: ",arm3*R[2])
plt.bar(["Arm 1","Arm 2","Arm 3"],R,color='blue')
plt.title("Initial Rewards")
plt.xlabel("Arms")
plt.ylabel("Rewards")
plt.show()
new_R=[arm1*R[0],arm2*R[1],arm3*R[2]]
plt.bar(["Arm 1","Arm 2","Arm 3"],new_R,color='blue')
plt.title("Rewards after iterations")
plt.xlabel("Arms")
plt.ylabel("Rewards")
plt.show()

# print(bounds)
# print(Q)

plt.plot(Q[0])
plt.plot(Q[1])
plt.plot(Q[2])
plt.show()
plt.pie(new_R,labels=["Arm 1","Arm 2","Arm 3"])
plt.show()

# plt.plot(total)
```

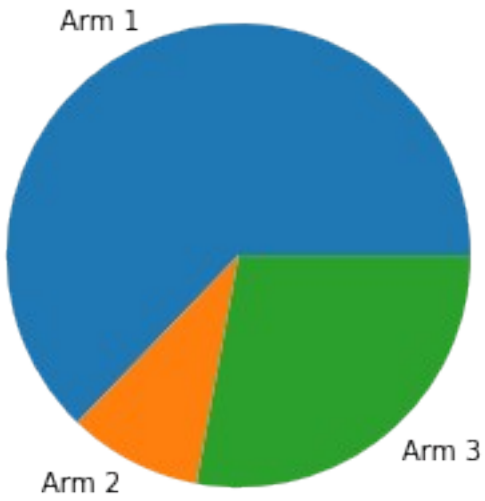
```
import matplotlib.pyplot as plt

bandit(R,0.3,100)
```

```
[[0], [0], [0]]
Number of times arms are selected:
54 16 30
The bound of a is:  0.7947711014267747
The bound of b is:  0.8007939399862771
The bound of c is:  0.797611817282866
Rewards for a are:  27.0
Rewards for b are:  4.0
Rewards for c are:  12.0

<ipython-input-61-e7c715b609ad>:40: RuntimeWarning: invalid value
encountered in double_scalars
  bounda=new_Qka+c*pow(np.log(i)/arm1,0.5)
<ipython-input-61-e7c715b609ad>:41: RuntimeWarning: invalid value
encountered in double_scalars
  boundb=new_Qkb+c*pow(np.log(i)/arm2,0.5)
<ipython-input-61-e7c715b609ad>:42: RuntimeWarning: invalid value
encountered in double_scalars
  boundc=new_Qkc+c*pow(np.log(i)/arm3,0.5)
<ipython-input-61-e7c715b609ad>:41: RuntimeWarning: divide by zero
encountered in double_scalars
  boundb=new_Qkb+c*pow(np.log(i)/arm2,0.5)
<ipython-input-61-e7c715b609ad>:42: RuntimeWarning: divide by zero
encountered in double_scalars
  boundc=new_Qkc+c*pow(np.log(i)/arm3,0.5)
```


Initial Rewards
Rewards
0.0
0.1
0.2
0.3
0.4
0.5
Arm 1
Arm 2
Arm 3
Arms


Rewards after iterations
Rewards
0
5
10
15
20
25
Arm 1
Arm 2
Arm 3
Arms


0.5
0.4
0.3
0.2
0.1
0.0
0
10
20
30
40
50


Arm 1
Arm 2
Arm 3