
IDS 435 - Assignment 5

- First Name: Yelizaveta
 - Last Name: Semikina
 - UIN: 670246811
-

ETC and UCB

This assignment is based on the slide deck used in the lectures titled "ETCvsUCB". Please refer to those slides for notation and algorithms.

Experiment Setup

Consider the following two-armed bandit setting: (i) you have in total $K = 2$ arms, that is, $a = 1$ or $a = 2$; (ii) each time that you play the first arm, you receive a reward that comes from a mean-zero normal distribution with a standard deviation of one; and (iii) for the second arm, this normal distribution has the mean of $-\delta$ and has the same standard deviation of one. Here, δ is a positive constant. Therefore, arm $a = 1$ is optimal and we have the gaps $\Delta_1 = 0$ and $\Delta_2 = \delta$ associated with arms $a = 1$ and $a = 2$, respectively. Indeed, we do not know the reward distributions, their means, or standard deviations when using the algorithm. We only have access to a sample from the reward distribution of an arm if it is chosen.

Below is code that illustrates how to compute regret for a simple rule that plays the first arm in odd rounds and the second one in even rounds. We have chosen $T = 1,000$ rounds when $\delta = .5$.

Questions

1. Suppose we set $T = 1,000$. If you rerun function `simple_rule` above, do you see the same regret? Compute the average regret by rerunning code `simple_rule(T=1000, delta=.5)` for 10^5 times. Do you see this average regret being close to $\delta \times N_t(a)$, where $N_t(a)$ is the number of times that action $a = 2$ has been selected by `simple_rule` prior to time t ? Explain your observation.
2. Write a python function implementing ETC. Note that inside your ETC function, you can simulate the reward you would receive by pulling an arm by calling the function `reward` for the pulled arm. Your ETC function will choose arms differently from the simple rule above.

3. Write a python function implementing UCB.
4. Plot a graph showing how the regret of ETC varies with δ . Specifically, on the x -axis vary δ between 0 and 1 in increments of 0.1, and on the y -axis, show the average of ETC regret over 10^5 reruns. When implementing ETC, set $T = 1000$ and use the optimal choice of m discussed on Page 6 of slides.
5. Fix $\delta = 0.1$ and $T = 2,000$. Plot a graph showing how the ETC regret varies with m . In other words, we want to understand how the behavior of ETC changes when using suboptimal choices for m . On the x -axis vary m between 0 and 400 in increments of 50, and on the y -axis, show the average of ETC regret over 10^5 reruns.
6. Plot the regret of UCB as a function of δ in a figure. Specifically, on the x -axis vary δ between 0 and 1 in increments of 0.1, and on the y -axis, show the average of UCB regret over 10^5 reruns. On the same figure, add a line showing how the regret of ETC with $m = 25$ varies as a function of δ . Add four more lines on the same figure for ETC with $m = 50, 75, 100$, and the optimal choice of m discussed on Page 6 of slides. Based on the results in the figure, how does the relative performance of UCB and ETC vary with m ? Further, comment on the importance of tuning m if one uses ETC?

In [1]:

```
import numpy as np

def reward(arm:int,delta:float):
    assert arm in [1,2]
    assert delta > 0
    if arm == 1:
        return np.random.normal(loc= 0,scale=1)
    else:
        return np.random.normal(loc= -delta,scale=1)

def simple_rule(T:int,delta:float):
    reward_arm_1 = []
    reward_arm_2 = []
    for t in range(1, T+1):
        if t%2 ==1:
            #play arm 1
            reward_arm_1.append(reward(1,delta))
        else:
            #play arm 2
            reward_arm_2.append(reward(2,delta))

    return reward_arm_1,reward_arm_2

if __name__ == "__main__":
    delta = .5
    T = 1000
    reward_arm_1,reward_arm_2 = simple_rule(T, delta)
    print('Sum of rewards from arm 1:\t', sum(reward_arm_1))
```

```

print('Sum of rewards from arm 2:\t', sum(reward_arm_2))
print('Observed regret:\t', sum(reward_arm_1)-sum(reward_arm_2))

print('=====')
# get expected regret
expected_regret = delta * (T / 2)

# calc observed regret
observed_regret = []
for i in range(1000):
    reward_arm_1, reward_arm_2 = simple_rule(T, delta)
    regret = sum(reward_arm_1) - sum(reward_arm_2)
    observed_regret.append(regret)

# calc average regret
avg_regret = sum(observed_regret) / len(observed_regret)

print("Expected regret:", expected_regret)
print("Average observed regret:", avg_regret)

```

```

Sum of rewards from arm 1:      -23.52547072407416
Sum of rewards from arm 2:      -206.0340152783841
Observed regret:                182.50854455430994
=====
Expected regret: 250.0
Average observed regret: 250.55522803492283

```

If you rerun function simple_rule above, do you see the same regret? Answer: No, because its random since the rewards generated by the function reward are random

In [2]:

```

import numpy as np

def simple_rule(T, delta):
    reward_arm_1 = []
    reward_arm_2 = []
    num_plays_arm_2 = 0
    regret = []
    estimated_regret = []

    for t in range(T):
        if t == 0:
            reward_arm_1.append(reward(1, delta))
            reward_arm_2.append(reward(2, delta))
            regret.append(reward_arm_2[-1] - reward_arm_1[-1])
            estimated_regret.append(regret[-1])
        else:
            if np.random.rand() < delta:
                # play arm 2
                reward_arm_2.append(reward(2, delta))
                num_plays_arm_2 += 1
                regret.append(reward_arm_1[-1] - reward_arm_2[-1])
                estimated_regret.append(estimated_regret[-1] + delta * (1 - esti
            else:
                # play arm 1
                reward_arm_1.append(reward(1, delta))
                regret.append(reward_arm_2[-1] - reward_arm_1[-1])
                estimated_regret.append(estimated_regret[-1] + (1 - delta) * (0

    return reward_arm_1, reward_arm_2, np.cumsum(regret), np.cumsum(estimated_re

```

```

if __name__ == "__main__":
    delta_ = .5
    num_runs = 5 # number of times to run the simulation
    T = 1000
    avg_regret = np.zeros(T)
    avg_est_regret = np.zeros(T)
    avg_num_plays_arm_2 = 0

    for i in range(num_runs):
        reward_arm_1, reward_arm_2, run_regret, run_est_regret, num_plays_arm_2 = si
        avg_regret += run_regret
        avg_est_regret += run_est_regret
        avg_num_plays_arm_2 += num_plays_arm_2

    avg_regret /= num_runs
    avg_est_regret /= num_runs
    avg_num_plays_arm_2 /= num_runs

    print('Average regret over', num_runs, 'runs:\n', avg_regret)
    print('Average estimated regret over', num_runs, 'runs:\n', avg_est_regret)
    print('Average number of times arm 2 was played over', num_runs, 'runs:\n',

```

Average regret over 5 runs:

```

[ -0.45485802 -0.89336429 -0.24634884 -1.50483587 -1.5146116
 -0.6066279 -0.15131186 -0.32138886  0.17967531  0.2847358
  0.17933258  0.40195069  0.51065255  0.60917533  0.88012858
  0.46436051  0.77740116  0.10919511 -1.12127935 -1.04804851
 -0.80168132 -1.24854507 -1.29819185 -1.32419916 -1.44411026
 -2.0877077 -2.26773312 -3.03806478 -3.05120523 -3.75980536
 -4.52129939 -4.23106297 -4.06485684 -4.34534803 -4.77610677
 -4.07447187 -3.86837107 -4.2937243 -4.67189222 -4.51508231
 -4.93964039 -5.15473303 -4.86331808 -4.55573303 -4.96499339
 -5.63608637 -4.23188594 -3.81184281 -4.32556603 -3.99220748
 -3.31930604 -2.92304083 -2.56845625 -1.92220617 -1.59851926
 -2.60483435 -2.79258248 -2.8036653 -1.63884544 -0.84497803
 -1.56836979 -2.38781148 -2.65845777 -2.30746569 -2.21593463
 -1.91280629 -1.92959851 -1.67660624 -1.5506698 -1.16064427
 -1.5107942 -1.29874731 -0.7408694 -1.17517214  0.2099244
  0.20318285  0.72145909  0.24690205  0.21801459 -0.89015344
 -1.06524582 -0.95560477 -2.26376854 -1.46449804 -2.47027437
 -2.88321853 -3.39490005 -4.04337363 -4.13855047 -5.28329751
 -4.50064463 -5.03475979 -6.17869446 -6.79885769 -7.91849881
 -7.46214177 -7.76307975 -8.65817088 -8.75932378 -8.39870338
 -9.47960408 -9.27622956 -9.67427937 -9.04682779 -8.7656964
 -9.31239053 -9.4785154 -9.95075156 -10.53311353 -10.61277793
 -10.10562024 -11.11757767 -9.89534144 -10.18171748 -10.0086538
 -10.56767544 -9.78800877 -9.9224869 -10.22624712 -9.61563161
 -9.37318542 -8.85470867 -9.61765037 -9.35118984 -9.06844465
 -8.77648404 -8.83029024 -9.18202575 -10.21201294 -10.76347239
 -11.02936617 -11.60965997 -11.68029394 -11.96682385 -11.08085893
 -10.32700252 -11.21974629 -10.4770987 -10.52821839 -10.54447662
 -11.03058385 -11.95515658 -12.98080413 -12.68862062 -13.90039484
 -12.58221608 -12.96245377 -13.24250711 -12.39847095 -12.3397165
 -12.3794564 -12.45613213 -13.58589325 -14.15575737 -14.14348948
 -15.18984402 -15.44177361 -14.74320163 -14.62521182 -14.88785858
 -15.252239 -14.44153519 -16.09937782 -15.75608398 -15.54142538

```

-16.30442186	-16.53491392	-16.47169419	-17.11324502	-18.23325718
-19.49497632	-20.31282178	-21.11220027	-20.92451755	-21.14053616
-20.54358736	-20.40530229	-21.81325283	-21.10756247	-20.78532295
-19.73456745	-19.73877654	-20.93502749	-20.71417553	-21.03752548
-21.42978136	-22.72197346	-22.91141856	-23.57419426	-22.92251285
-24.27498822	-24.62157673	-24.98623476	-24.43025029	-25.49826609
-26.68234146	-28.95375092	-28.7337104	-29.1710712	-28.81185434
-28.33958243	-28.93953466	-29.33575618	-29.49549082	-29.70545546
-29.74307302	-29.32170468	-29.88015715	-30.49677694	-30.72704833
-29.32074726	-29.46314856	-29.71873236	-29.83387253	-30.60856571
-30.09803919	-30.9891339	-30.80913265	-32.32126249	-34.04853594
-33.80067157	-35.19829435	-34.29177149	-34.11165749	-33.4937903
-32.60123755	-32.39524676	-31.97576547	-31.5566419	-31.79166066
-32.56583857	-33.35729308	-32.96702735	-33.14163889	-32.69667237
-32.83857368	-33.873932	-34.26307732	-34.53719691	-33.54232843
-34.54623651	-33.42107636	-32.88248462	-34.28819282	-33.02146318
-34.30876944	-35.75347345	-35.90733694	-36.66898435	-36.65065868
-37.33628846	-37.36038707	-38.47116089	-39.12394614	-38.87830412
-38.315146	-38.31455535	-39.20723244	-39.22781123	-38.45499327
-39.27807661	-38.89859846	-39.3950747	-39.88437817	-41.2244209
-42.68903238	-42.51733768	-43.10447676	-43.52570047	-44.59910883
-45.17199336	-45.07440607	-46.09715045	-46.20816301	-46.10970393
-46.16447732	-45.64850103	-45.50980112	-45.12422366	-43.63141368
-44.36558033	-45.3253546	-45.11859997	-46.10570622	-45.98011878
-46.08292503	-46.60797978	-47.32242326	-46.66309417	-47.18062672
-47.61151002	-46.62259731	-47.18237957	-47.36872713	-47.84793632
-47.15000636	-47.15523748	-47.8186141	-48.20913811	-47.66937141
-47.87042221	-48.56777473	-48.86949804	-48.17853963	-47.00578541
-48.13245176	-46.77834396	-47.37042922	-48.08454478	-47.89885146
-48.0878556	-47.45724877	-48.64706501	-48.41031964	-47.99494121
-47.31820236	-48.97999391	-48.11971989	-47.79750953	-47.48193704
-46.30440533	-45.45935015	-47.12883788	-45.87760198	-46.94898339
-47.11707953	-47.44842592	-46.31131506	-45.86714609	-45.59855614
-45.1149085	-46.0099258	-46.40717185	-45.9379	-45.61022097
-45.69741053	-45.67270669	-47.22145582	-46.92371814	-47.05598469
-48.06154758	-47.71602035	-47.65184889	-47.68249103	-46.64625361
-46.45204557	-47.25432462	-46.68031643	-46.72422487	-48.45383621
-48.56952724	-50.10334799	-50.30634814	-51.14218445	-50.79897966
-49.99819497	-49.66603182	-49.46680322	-50.1727316	-50.0786739
-51.09667751	-52.03214841	-52.11313661	-52.25134498	-51.55682886
-53.06457355	-53.58927699	-54.10894395	-53.34285121	-53.81969462
-53.32650899	-54.38403295	-54.6514216	-54.9486532	-54.77134156
-54.57627378	-55.99334453	-56.04798959	-55.8819145	-55.72125337
-55.33160159	-55.90062181	-56.20258976	-55.98158858	-55.08369002
-54.94498899	-54.11330303	-55.01036456	-54.69330941	-54.84427765
-54.8125677	-54.3947216	-53.82395234	-53.06007339	-53.69670771
-54.43863395	-55.00092432	-55.12553751	-55.74825151	-57.30896962
-57.38815895	-58.18861597	-58.17910649	-58.70108458	-57.91845921
-58.39080479	-58.69111532	-58.57996427	-58.77529745	-59.31049187
-58.84396663	-58.35080576	-58.12229552	-58.43694552	-58.2222588
-58.45324981	-58.29613505	-57.71535899	-57.68829521	-58.84290309
-57.82019124	-56.86510074	-57.12823332	-57.52576861	-58.62012853
-59.81689197	-60.59108567	-60.17568382	-59.57304159	-59.05982571
-59.3927998	-58.51501697	-58.82204733	-58.99384911	-58.88732844
-58.47569942	-60.16927819	-59.92354346	-60.21266682	-60.50254733
-60.19763604	-59.40135742	-59.50462862	-59.82827192	-61.00911923
-59.13043901	-57.09620238	-56.9907163	-56.33287652	-55.83117318
-54.00467446	-54.44529387	-53.61194155	-53.59770516	-53.36543659
-52.73418705	-52.45943721	-52.33775483	-52.60089696	-52.41857991
-52.43560937	-52.41164442	-52.77168941	-53.62757387	-52.49707509

-52.8459028	-52.69631734	-53.76707109	-54.71047375	-53.75240431
-53.99723804	-53.72559044	-53.44172038	-53.7371352	-54.17006857
-54.16428427	-54.64241961	-54.31208133	-53.13253686	-54.47097296
-53.78554122	-53.95779891	-54.6776187	-55.36642651	-55.3469614
-54.3286609	-54.65586136	-53.88573615	-53.93798901	-54.15038289
-54.88487478	-54.27845997	-54.49025163	-54.99144776	-55.64924354
-55.18468748	-55.30746232	-56.08449286	-56.38643591	-58.2406219
-59.08821758	-59.82228581	-61.20098033	-62.0052369	-62.58222724
-61.53973404	-61.49589332	-61.00278706	-60.68444646	-61.03195157
-60.58250329	-62.01177774	-61.05698518	-60.44028589	-59.52108253
-58.51931184	-59.7023017	-58.87830944	-57.73597818	-56.84334219
-56.63699865	-56.96715838	-55.59618631	-56.13470256	-55.99262822
-55.97677109	-57.83612666	-59.36989572	-58.371943	-58.71295745
-58.91542265	-59.10611399	-59.50742284	-61.02603318	-61.90768014
-63.20335718	-63.36062402	-64.68664529	-65.23397547	-63.80047484
-65.47923254	-65.55217439	-64.84529615	-64.27437922	-65.28354284
-65.47093459	-64.74629198	-64.65460406	-63.8298222	-64.2017553
-64.72401298	-64.47039522	-64.28980076	-64.3677432	-63.47294235
-63.13695043	-64.23792269	-65.36003783	-65.22179657	-66.78779733
-67.66186634	-67.56927933	-68.11421928	-68.58278949	-68.79215925
-68.1772681	-69.03869335	-69.22134897	-70.09283904	-70.65438929
-71.27333517	-71.43788649	-70.71171574	-71.34597555	-71.28562808
-71.61986414	-71.85417911	-72.01808229	-72.68354143	-72.0033466
-72.76936817	-73.63467489	-73.25160033	-74.46911245	-74.80324042
-74.3464656	-73.810247	-73.47578378	-72.78993357	-73.81460525
-73.56692201	-73.36976399	-73.45671721	-73.71818408	-74.48639443
-73.74202417	-73.55606457	-74.17689862	-73.80933865	-73.21156921
-72.15460184	-71.85166187	-72.1973323	-72.96686583	-72.78833963
-72.60498802	-71.32963493	-71.15218022	-71.26992569	-70.79881509
-72.16773018	-71.97011436	-72.52067111	-72.22087981	-72.23901831
-72.85481061	-72.19327495	-72.59267486	-72.37526894	-72.05560838
-71.27322169	-70.37850082	-71.80958865	-72.98870677	-72.86415153
-72.34962337	-71.78977837	-72.00545335	-72.05930942	-73.23513242
-73.92468938	-73.8383363	-73.10878696	-72.75417975	-72.5182301
-73.09072796	-74.27958552	-74.83194153	-74.84097807	-74.73799159
-74.48380916	-74.09362524	-72.57147378	-72.05359451	-72.49509629
-72.88736403	-72.50207606	-72.35021015	-72.54532615	-72.06675428
-72.20427199	-72.38649812	-72.07877205	-72.02945848	-71.48364921
-72.04255593	-72.62861729	-71.8048492	-72.55281191	-72.30555446
-72.37568011	-71.50095856	-71.32613754	-71.5245466	-71.0179023
-70.24272229	-71.23624536	-72.05621926	-72.53186336	-72.11155949
-73.33914505	-73.74176456	-74.45285867	-73.44498604	-73.2326707
-74.23931938	-74.65584991	-74.84366106	-75.15388319	-75.51808827
-76.07736703	-75.05806274	-74.89015105	-75.45469125	-76.24686275
-76.64253818	-77.20059582	-76.63236704	-76.52316123	-76.77218401
-76.9500297	-76.06355995	-75.13400102	-74.1698855	-74.68674005
-75.78167273	-76.52619659	-77.95124635	-77.27105051	-77.02154342
-77.14953366	-77.63596673	-77.893469	-78.84153251	-78.46470297
-78.30485639	-78.76220614	-77.92188512	-78.13492775	-78.20159876
-77.92001404	-77.94718175	-77.93699998	-78.11637892	-78.80514095
-78.06467513	-77.82314287	-78.89443646	-78.26358649	-78.56540707
-77.54626292	-78.41954744	-77.7376263	-77.71522308	-77.3933101
-77.16548665	-78.63125678	-78.08239549	-77.88632426	-77.48099644
-76.75364508	-77.25509893	-77.77500701	-77.8249819	-77.80288795
-77.37005127	-77.01706897	-76.08727865	-76.75573857	-76.37523846
-76.17237969	-76.98417561	-78.48978915	-78.98369871	-79.38796437
-78.44941485	-78.77262353	-78.71431089	-77.96951074	-76.91368553
-78.22667272	-78.49829155	-78.20654421	-78.3268766	-79.47006973
-79.99708253	-79.98569043	-79.12315362	-79.40512769	-78.45752165
-78.32303747	-78.19879148	-79.12258659	-77.45438696	-77.55406446

```

-76.75523547 -76.82346524 -76.01413294 -75.2656067 -75.69209868
-74.78458897 -75.26287918 -74.8194166 -75.70738127 -75.80548194
-76.28723883 -77.45450196 -76.95083583 -77.18736271 -77.33508238
-77.05247185 -77.95512548 -78.12993915 -78.16271084 -79.31162735
-78.91649927 -78.81686751 -77.89354428 -77.71920657 -78.20140574
-78.77912838 -78.0070833 -77.7295938 -77.08899134 -76.50028913
-75.74938006 -76.04196573 -75.75091296 -76.35229321 -76.80649395
-77.41530706 -77.18329763 -78.70383511 -79.56242915 -80.025304
-80.71005669 -80.18118902 -80.02029095 -80.9595702 -81.41536628
-81.68433807 -81.62413553 -81.47919501 -81.10308619 -80.39702937
-79.68775392 -80.10796734 -79.230545 -79.94246398 -79.62134268
-79.45853904 -80.4861295 -79.81298745 -79.96945549 -80.28278213
-80.3917209 -80.9773991 -79.47947972 -80.05703977 -79.1412764
-79.21737531 -79.81230194 -79.91404524 -79.5081164 -79.17002768
-80.65823764 -80.8335433 -81.06220158 -81.25443041 -83.40594835
-83.52829771 -83.8620712 -83.66145996 -83.71215411 -84.19920636
-83.93207313 -83.5379231 -84.11240685 -84.81045016 -85.17905344
-84.96914466 -84.00998804 -83.70269272 -84.16317555 -85.57348751
-84.76588631 -85.09112823 -84.27726314 -84.58949891 -83.05050286
-83.23972039 -82.99724841 -83.09026835 -83.09917468 -83.14380765
-83.14256485 -83.52353606 -84.42556815 -83.63208627 -84.42794281
-83.8780802 -84.49306432 -84.12398666 -84.60919064 -84.94103807
-84.92772541 -84.69969086 -84.28496329 -84.92415622 -83.96607393
-83.68207622 -83.73883504 -83.54418816 -84.4162615 -84.44263698
-84.68865925 -84.77020541 -84.89128014 -85.30856097 -85.69715358
-87.14471636 -87.16638875 -87.03068711 -87.48900508 -88.12536926
-87.89398255 -87.55341903 -87.04813063 -87.29381527 -87.55693138
-87.68282608 -87.73293963 -88.17976615 -88.91305814 -88.63696408
-87.65214136 -87.97398685 -88.14703683 -89.17900229 -89.47604371
-88.90943252 -89.40396351 -89.65772721 -88.98246097 -89.42877792
-89.14554586 -87.87413579 -88.52782253 -88.26773732 -89.03654377
-88.7741417 -88.54385578 -88.45476812 -89.44247495 -89.09494661
-89.34158588 -89.0981365 -88.96200269 -89.58812463 -90.433642
-90.7623965 -91.36659843 -90.67012836 -90.41537124 -89.98344322
-90.97703556 -90.59414989 -90.87914596 -90.47652816 -90.98353903
-92.23202717 -91.72950968 -91.57359231 -90.99874177 -90.28584396
-89.82183762 -90.76895396 -90.573783 -91.31382246 -91.19635857
-91.8746771 -91.11344523 -91.1071584 -90.99785485 -91.43609319
-92.08523596 -91.83131258 -91.68951744 -91.95236202 -91.72263611
-92.16065458 -91.64224124 -90.99832901 -91.53852564 -92.24775475
-92.23456297 -92.45882472 -92.87185649 -92.6140064 -93.45019929
-93.50008852 -93.87707232 -92.67761327 -92.68872108 -92.64042188
-93.16697374 -94.08841576 -94.64999091 -94.46577807 -94.36710561
-93.633987 -93.84305471 -93.39870875 -93.93758687 -94.13885844
-94.96769058 -94.5723601 -93.45181702 -93.46658884 -93.95284243
-93.73859426 -93.63287516 -93.78433608 -93.7343809 -92.69877689
-92.48228128 -92.03346083 -91.76785685 -90.1962514 -91.75974928]

```

Average estimated regret over 5 runs:

```

[-4.54858015e-01 -3.82287023e-01 -4.60015265e-02 3.22141222e-01
8.06212596e-01 1.34824828e+00 2.01926613e+00 2.45477505e+00
2.87252951e+00 3.18140674e+00 3.63584535e+00 4.16306466e+00
4.62667432e+00 5.05847914e+00 5.57438156e+00 6.23233276e+00
6.86130837e+00 7.47579617e+00 7.78304007e+00 8.03666202e+00
8.26347299e+00 8.77687848e+00 9.23358123e+00 9.76193260e+00
1.01261083e+01 1.04081961e+01 1.08492400e+01 1.12697620e+01
1.17800230e+01 1.22351535e+01 1.25627187e+01 1.28265013e+01
1.34583927e+01 1.39743383e+01 1.45323111e+01 1.51112976e+01
1.55007908e+01 1.56955374e+01 1.59929107e+01 1.64415973e+01
1.66659406e+01 1.71781123e+01 1.76341981e+01 1.82622411e+01
1.87762625e+01 1.92332732e+01 1.96617786e+01 2.01760313e+01

```

2.07331576e+01	2.13117208e+01	2.18010024e+01	2.22456432e+01
2.27679636e+01	2.34291238e+01	2.39597039e+01	2.45249939e+01
2.50076389e+01	2.55489615e+01	2.62196227e+01	2.68549533e+01
2.71726187e+01	2.74314513e+01	2.76608676e+01	2.81755758e+01
2.86329299e+01	2.91616069e+01	2.96259454e+01	3.01581147e+01
3.05241993e+01	3.08072417e+01	3.12487628e+01	3.16695234e+01
3.20799037e+01	3.24850938e+01	3.28876889e+01	3.32889864e+01
3.37896352e+01	3.43399596e+01	3.47151218e+01	3.53027029e+01
3.59964934e+01	3.66433887e+01	3.72668363e+01	3.79785602e+01
3.85344221e+01	3.90123530e+01	3.93513185e+01	3.96208012e+01
3.98555426e+01	4.01729133e+01	4.06315986e+01	4.10609413e+01
4.14756126e+01	4.18829483e+01	4.20866161e+01	4.25884501e+01
4.32393670e+01	4.36648255e+01	4.42775547e+01	4.47839193e+01
4.51371017e+01	4.56136928e+01	4.61519884e+01	4.65211362e+01
4.70057101e+01	4.73479970e+01	4.75191405e+01	4.80047122e+01
4.84474981e+01	4.90688910e+01	4.94795875e+01	4.98849357e+01
5.04876099e+01	5.08889469e+01	5.13896154e+01	5.17399497e+01
5.22151168e+01	5.26527004e+01	5.29714922e+01	5.33308881e+01
5.37105860e+01	5.42004350e+01	5.44453595e+01	5.49678217e+01
5.54290528e+01	5.61596684e+01	5.67249762e+01	5.72076301e+01
5.75489570e+01	5.78196205e+01	5.81549522e+01	5.85226181e+01
5.92064510e+01	5.96483675e+01	6.02693257e+01	6.09798049e+01
6.14350444e+01	6.19626642e+01	6.23264741e+01	6.28083790e+01
6.31493315e+01	6.35198077e+01	6.40050459e+01	6.45476649e+01
6.51189744e+01	6.56046292e+01	6.59474566e+01	6.61188703e+01
6.66045771e+01	6.72474305e+01	6.78688573e+01	6.82795706e+01
6.86849273e+01	6.90876056e+01	6.92889448e+01	6.95896144e+01
7.01399492e+01	7.06151166e+01	7.12527003e+01	7.17714921e+01
7.22308880e+01	7.27605860e+01	7.31254350e+01	7.35078595e+01
7.37990717e+01	7.40446778e+01	7.42674809e+01	7.45788824e+01
7.49345832e+01	7.52124336e+01	7.54513588e+01	7.55708214e+01
7.58305527e+01	7.60604183e+01	7.63753511e+01	7.69328176e+01
7.73115508e+01	7.77009174e+01	7.80956007e+01	7.84929423e+01
7.89916131e+01	7.96409486e+01	8.03656163e+01	8.10279501e+01
8.18591170e+01	8.25747005e+01	8.30324922e+01	8.34613881e+01
8.37758360e+01	8.43330600e+01	8.48116720e+01	8.53509780e+01
8.59206310e+01	8.63054575e+01	8.67978707e+01	8.71440773e+01
8.75171807e+01	8.78037323e+01	8.79470081e+01	8.85186461e+01
8.90044650e+01	8.93473745e+01	8.97188292e+01	9.01045566e+01
9.06974203e+01	9.12938521e+01	9.18920680e+01	9.24911760e+01
9.27907300e+01	9.30405070e+01	9.33653955e+01	9.38278397e+01
9.44590618e+01	9.50746729e+01	9.55824784e+01	9.63363812e+01
9.71133326e+01	9.78018083e+01	9.82460461e+01	9.86681650e+01
9.90792245e+01	9.95847542e+01	1.00137519e+02	1.00713902e+02
1.01302093e+02	1.01996188e+02	1.02543236e+02	1.03116760e+02
1.03603522e+02	1.04146903e+02	1.04618593e+02	1.05054439e+02
1.05572361e+02	1.06231323e+02	1.06860803e+02	1.07375544e+02
1.08032914e+02	1.08661599e+02	1.09175941e+02	1.09733113e+02
1.10311698e+02	1.10800991e+02	1.11345638e+02	1.11717961e+02
1.12204122e+02	1.12547203e+02	1.12818744e+02	1.13254514e+02
1.13672399e+02	1.13981341e+02	1.14335813e+02	1.14813048e+02
1.15351666e+02	1.15820975e+02	1.16255630e+02	1.16672957e+02
1.17181620e+02	1.17735952e+02	1.18013118e+02	1.18451701e+02
1.18770992e+02	1.19230638e+02	1.19560461e+02	1.20025373e+02
1.20457828e+02	1.20874056e+02	1.21382170e+02	1.21836227e+02
1.22363255e+02	1.22826770e+02	1.23258527e+02	1.23774405e+02
1.24232345e+02	1.24861314e+02	1.25375799e+02	1.25733042e+02
1.26111663e+02	1.26700973e+02	1.27195629e+02	1.27642956e+02
1.28066620e+02	1.28578452e+02	1.29134368e+02	1.29412326e+02
1.29751305e+02	1.30220794e+02	1.30755539e+02	1.31222912e+02

1.31756598e+02	1.32323441e+02	1.32806862e+02	1.33548573e+02
1.34119429e+02	1.34704856e+02	1.35297570e+02	1.35993927e+02
1.36342106e+02	1.36816195e+02	1.37053239e+02	1.37271762e+02
1.37581023e+02	1.37835653e+02	1.38162969e+02	1.38626626e+02
1.38958455e+02	1.39324370e+02	1.39807327e+02	1.40248805e+02
1.40869545e+02	1.41479914e+02	1.42085099e+02	1.42887692e+02
1.43388988e+02	1.44039636e+02	1.44564960e+02	1.45127622e+02
1.45508953e+02	1.46099618e+02	1.46694951e+02	1.47392618e+02
1.47941451e+02	1.48715867e+02	1.49203076e+02	1.49746680e+02
1.50218482e+02	1.50654383e+02	1.51072333e+02	1.51681309e+02
1.52285796e+02	1.52888040e+02	1.53589162e+02	1.54039723e+02
1.54565003e+02	1.55227644e+02	1.55858964e+02	1.56374624e+02
1.56732454e+02	1.57011369e+02	1.57350826e+02	1.57920555e+02
1.58505420e+02	1.59197852e+02	1.59744068e+02	1.60217176e+02
1.60953730e+02	1.61722007e+02	1.62306145e+02	1.62898215e+02
1.63394249e+02	1.63742267e+02	1.64116275e+02	1.64503280e+02
1.64796782e+02	1.65043533e+02	1.65466908e+02	1.65878596e+02
1.66284440e+02	1.66787362e+02	1.67338823e+02	1.67814553e+02
1.68352419e+02	1.68921351e+02	1.69505818e+02	1.70098051e+02
1.70394167e+02	1.70942226e+02	1.71416255e+02	1.71853269e+02
1.72471777e+02	1.72881030e+02	1.73385657e+02	1.73837971e+02
1.74364127e+02	1.74727206e+02	1.75108745e+02	1.75399514e+02
1.75844899e+02	1.76367592e+02	1.76928938e+02	1.77509611e+02
1.78199947e+02	1.78845116e+02	1.79467700e+02	1.79978992e+02
1.80434638e+02	1.80862461e+02	1.81276372e+02	1.81783328e+02
1.82336806e+02	1.82813545e+02	1.83351915e+02	1.83921099e+02
1.84405692e+02	1.84947988e+02	1.85519136e+02	1.85904710e+02
1.86297497e+02	1.86693890e+02	1.87192087e+02	1.87541186e+02
1.88015735e+02	1.88353009e+02	1.88621647e+02	1.89155965e+02
1.89823125e+02	1.90356704e+02	1.90923494e+02	1.91506889e+02
1.91998587e+02	1.92644435e+02	1.93167360e+02	1.93528822e+02
1.93909553e+02	1.94299918e+02	1.94795101e+02	1.95442693e+02
1.96066488e+02	1.96678386e+02	1.96984335e+02	1.97337310e+02
1.97713797e+02	1.98102040e+02	1.98396162e+02	1.98843223e+02
1.99366754e+02	1.99928519e+02	2.00609401e+02	2.01149843e+02
2.01720063e+02	2.02405174e+02	2.03047729e+02	2.03769006e+02
2.04429645e+02	2.04959965e+02	2.05525124e+02	2.06107704e+02
2.06598994e+02	2.07144639e+02	2.07617461e+02	2.08053873e+02
2.08472078e+02	2.08981181e+02	2.09535733e+02	2.09813008e+02
2.10251646e+02	2.10770965e+02	2.11430625e+02	2.12060454e+02
2.12575369e+02	2.13132827e+02	2.13811555e+02	2.14250920e+02
2.14870602e+02	2.15380443e+02	2.15835363e+02	2.16262824e+02
2.16876554e+02	2.17483419e+02	2.18086851e+02	2.18688568e+02
2.19289426e+02	2.19889855e+02	2.20590069e+02	2.21340177e+02
2.21915230e+02	2.22402757e+02	2.23046521e+02	2.23368402e+02
2.23629343e+02	2.24059814e+02	2.24575049e+02	2.25232666e+02
2.26061475e+02	2.26575880e+02	2.27033082e+02	2.27461683e+02
2.27675983e+02	2.27883134e+02	2.28286709e+02	2.28788496e+02
2.29239390e+02	2.29564837e+02	2.29927561e+02	2.30408922e+02
2.30749603e+02	2.31219944e+02	2.31655114e+02	2.32272699e+02
2.32881491e+02	2.33485888e+02	2.33888086e+02	2.34389185e+02
2.34939734e+02	2.35315009e+02	2.35802647e+02	2.36346465e+02
2.37118375e+02	2.37604329e+02	2.37947307e+02	2.38218795e+02
2.38354540e+02	2.38622412e+02	2.38956348e+02	2.39323316e+02
2.39806800e+02	2.40448542e+02	2.41169413e+02	2.41929848e+02
2.42510066e+02	2.43000175e+02	2.43445230e+02	2.43967757e+02
2.44329020e+02	2.44909652e+02	2.45499968e+02	2.45995126e+02
2.46442705e+02	2.46966494e+02	2.47328389e+02	2.47809337e+02
2.48249810e+02	2.48670047e+02	2.49180166e+02	2.49635225e+02
2.50162754e+02	2.50626519e+02	2.50958402e+02	2.51224343e+02

2.51657313e+02	2.52073799e+02	2.52582041e+02	2.53136163e+02
2.53413223e+02	2.53751754e+02	2.54121019e+02	2.54705651e+02
2.55197968e+02	2.55744126e+02	2.56217205e+02	2.56853744e+02
2.57372014e+02	2.57831149e+02	2.58260717e+02	2.58775500e+02
2.59032892e+02	2.59261588e+02	2.59675936e+02	2.60083110e+02
2.60686697e+02	2.61188490e+02	2.61639387e+02	2.61964836e+02
2.62427560e+02	2.62858922e+02	2.63374603e+02	2.63832443e+02
2.64261364e+02	2.64575824e+02	2.64833054e+02	2.65161669e+02
2.65625976e+02	2.66158130e+02	2.66924207e+02	2.67407246e+02
2.67748765e+02	2.68219524e+02	2.68854904e+02	2.69572594e+02
2.70131439e+02	2.70610861e+02	2.71150573e+02	2.71420428e+02
2.71855356e+02	2.72272820e+02	2.72681552e+02	2.72985918e+02
2.73438101e+02	2.73964192e+02	2.74527238e+02	2.75008761e+02
2.75449523e+02	2.75869903e+02	2.76480094e+02	2.77085189e+02
2.77487736e+02	2.78089010e+02	2.78689647e+02	2.79389966e+02
2.79940125e+02	2.80415204e+02	2.80952744e+02	2.81421514e+02
2.81855899e+02	2.82273091e+02	2.82781688e+02	2.83135986e+02
2.83513135e+02	2.83901709e+02	2.84295997e+02	2.84793140e+02
2.85241712e+02	2.85765998e+02	2.86328141e+02	2.86809212e+02
2.87149748e+02	2.87720016e+02	2.88405150e+02	2.89147717e+02
2.89719000e+02	2.90304642e+02	2.90897463e+02	2.91693874e+02
2.92392079e+02	2.93041181e+02	2.93665733e+02	2.94378008e+02
2.95134146e+02	2.95612215e+02	2.96051250e+02	2.96670767e+02
2.97480525e+02	2.98285405e+02	2.98887844e+02	2.99289064e+02
2.99789674e+02	3.00239979e+02	3.00665131e+02	3.01077708e+02
3.01383996e+02	3.01637140e+02	3.01863712e+02	3.02176998e+02
3.02733641e+02	3.03411962e+02	3.04051123e+02	3.04670704e+02
3.05280494e+02	3.05885389e+02	3.06387836e+02	3.06839060e+02
3.07264672e+02	3.07777478e+02	3.08433881e+02	3.09062082e+02
3.09776183e+02	3.10433234e+02	3.11061759e+02	3.11676021e+02
3.12083153e+02	3.12286718e+02	3.12688501e+02	3.13289393e+02
3.13889838e+02	3.14590061e+02	3.15340173e+02	3.16015228e+02
3.16552756e+02	3.17221520e+02	3.17755902e+02	3.18523093e+02
3.19106688e+02	3.19698486e+02	3.20194385e+02	3.20542335e+02
3.20916309e+02	3.21403297e+02	3.21846790e+02	3.22368537e+02
3.22729411e+02	3.23209847e+02	3.23850066e+02	3.24470175e+02
3.24980229e+02	3.25535257e+02	3.26012770e+02	3.26451527e+02
3.26870906e+02	3.27480595e+02	3.28085439e+02	3.28787862e+02
3.29439073e+02	3.30164678e+02	3.30927481e+02	3.31508883e+02
3.32099583e+02	3.32794934e+02	3.33442609e+02	3.34066446e+02
3.34578365e+02	3.35134325e+02	3.35512304e+02	3.35801294e+02
3.36345789e+02	3.37118037e+02	3.37804160e+02	3.38447222e+02
3.38968753e+02	3.39529519e+02	3.40209901e+02	3.40950093e+02
3.41820188e+02	3.42555236e+02	3.43322760e+02	3.44106522e+02
3.44598403e+02	3.44944343e+02	3.45217314e+02	3.45453799e+02
3.46072041e+02	3.46581163e+02	3.47235723e+02	3.47663004e+02
3.48276644e+02	3.48983464e+02	3.49636874e+02	3.50363579e+02
3.50826931e+02	3.51258608e+02	3.51874446e+02	3.52482365e+02
3.53286324e+02	3.53788304e+02	3.54339294e+02	3.55014789e+02
3.55552536e+02	3.56221410e+02	3.56855847e+02	3.57673066e+02
3.58281675e+02	3.58785979e+02	3.59438132e+02	3.59964208e+02
3.60427246e+02	3.60658765e+02	3.61074524e+02	3.61582404e+02
3.62136344e+02	3.62713314e+02	3.63301799e+02	3.63996041e+02
3.64543163e+02	3.65016723e+02	3.65553504e+02	3.66121894e+02
3.66706089e+02	3.67398186e+02	3.68044235e+02	3.68567260e+02
3.69028772e+02	3.69459528e+02	3.69874906e+02	3.70282595e+02
3.70786439e+02	3.71438362e+02	3.71864323e+02	3.72077303e+02
3.72283794e+02	3.72687039e+02	3.73188661e+02	3.73739473e+02
3.74214878e+02	3.74752581e+02	3.75321433e+02	3.76005858e+02
3.76548071e+02	3.76919178e+02	3.77304731e+02	3.77897507e+02

3.78293896e+02	3.78692090e+02	3.79191187e+02	3.79640735e+02
3.80265510e+02	3.80777897e+02	3.81234090e+02	3.81562187e+02
3.82026236e+02	3.82358260e+02	3.82924272e+02	3.83307278e+02
3.83698781e+02	3.83994532e+02	3.84342408e+02	3.84916346e+02
3.85403315e+02	3.86046800e+02	3.86568542e+02	3.87129413e+02
3.87809848e+02	3.88550066e+02	3.89020175e+02	3.89655230e+02
3.90372757e+02	3.90931520e+02	3.91710902e+02	3.92500593e+02
3.92995439e+02	3.93342861e+02	3.93916573e+02	3.94503428e+02
3.95096856e+02	3.95493570e+02	3.95891927e+02	3.96291105e+02
3.96690695e+02	3.97190489e+02	3.97540387e+02	3.98015335e+02
3.98352810e+02	3.98721547e+02	3.99205915e+02	3.99648100e+02
4.00269192e+02	4.00979738e+02	4.01635011e+02	4.02262647e+02
4.02776466e+02	4.03433375e+02	4.03761829e+02	4.04126057e+02
4.04508170e+02	4.04899227e+02	4.05294756e+02	4.05792520e+02
4.06241402e+02	4.06765843e+02	4.07228063e+02	4.07859174e+02
4.08574729e+02	4.09132506e+02	4.09711395e+02	4.10500840e+02
4.10995562e+02	4.11542923e+02	4.12116603e+02	4.12603444e+02
4.13146864e+02	4.13518574e+02	4.14004429e+02	4.14247356e+02
4.14668820e+02	4.15179552e+02	4.15634918e+02	4.15962601e+02
4.16226442e+02	4.16558363e+02	4.17024324e+02	4.17357304e+02
4.17723794e+02	4.18207039e+02	4.18848661e+02	4.19469473e+02
4.19979878e+02	4.20435081e+02	4.20862683e+02	4.21376483e+02
4.21933384e+02	4.22511834e+02	4.23101059e+02	4.23695671e+02
4.24192978e+02	4.24741631e+02	4.25415957e+02	4.25953121e+02
4.26521702e+02	4.27205993e+02	4.27748139e+02	4.28319211e+02
4.29004748e+02	4.29547516e+02	4.30118900e+02	4.30604592e+02
4.31247438e+02	4.31868861e+02	4.32379572e+02	4.33034928e+02
4.33362606e+02	4.33826445e+02	4.34458365e+02	4.35274324e+02
4.36082304e+02	4.36586294e+02	4.37238289e+02	4.37764286e+02
4.38327285e+02	4.38908785e+02	4.39299534e+02	4.39894909e+02
4.40492597e+02	4.40891440e+02	4.41190862e+02	4.41540573e+02
4.41915429e+02	4.42302856e+02	4.42596570e+02	4.43043427e+02
4.43466856e+02	4.43678570e+02	4.43884427e+02	4.44387355e+02
4.44838820e+02	4.45264552e+02	4.45577418e+02	4.45933851e+02
4.46212067e+02	4.46551176e+02	4.47020730e+02	4.47455507e+02
4.47672895e+02	4.47981590e+02	4.48335937e+02	4.49013110e+02
4.49651697e+02	4.50170991e+02	4.50730637e+02	4.51310461e+02
4.51900372e+02	4.52295328e+02	4.52892806e+02	4.53591545e+02
4.53940914e+02	4.54315599e+02	4.54602942e+02	4.54846613e+02
4.55068448e+02	4.55479366e+02	4.56084825e+02	4.56587555e+02
4.57138919e+02	4.57614602e+02	4.58252443e+02	4.58871363e+02
4.59580824e+02	4.60135554e+02	4.60712919e+02	4.61001601e+02
4.61545943e+02	4.62018113e+02	4.62454199e+02	4.62972241e+02
4.63431263e+02	4.63860773e+02	4.64475529e+02	4.65082906e+02
4.65586595e+02	4.66038440e+02	4.66464362e+02	4.66777323e+02
4.67133803e+02	4.67512044e+02	4.68001164e+02	4.68345724e+02
4.68918004e+02	4.69304144e+02	4.69697214e+02	4.69993749e+02
4.70242016e+02	4.70666150e+02	4.70978217e+02	4.71334251e+02
4.71712267e+02	4.72201276e+02	4.72845780e+02	4.73368032e+02
4.73829158e+02	4.74459721e+02	4.74975002e+02	4.75632643e+02
4.76061464e+02	4.76575874e+02	4.77033079e+02	4.77461681e+02
4.78075983e+02	4.78683133e+02	4.79286709e+02	4.79788496e+02
4.80339390e+02	4.80914837e+02	4.81402561e+02	4.81846422e+02
4.82368353e+02	4.82729319e+02	4.83309801e+02	4.83800043e+02
4.84145163e+02	4.84517724e+02	4.85004004e+02	4.85547144e+02
4.85918714e+02	4.86404499e+02	4.86947391e+02	4.87318838e+02
4.87904561e+02	4.88497422e+02	4.89093853e+02	4.89592069e+02
4.89941176e+02	4.90315730e+02	4.90903007e+02	4.91196646e+02]

Average number of times arm 2 was played over 5 runs:

492.4

Do you see this average regret being close to $\delta \times N_t(a)$, where $N_t(a)$ is the number of times that action

a

2 has been selected by simple_rule prior to time t ? Explain your observation.

ANSWER: Looking at the output, we can see that the average number of times arm 2 was played over 5 runs is approximately 499.4. We can use this value to compute the estimated regret based on the formula $\delta \times N_t(a)$ for each time step t .

We can also see that the estimated regret values increase with t , as expected.

To determine whether the average regret is close to the estimated regret, we can compare the two values at each time step t . If they are close to each other, then we can say that the simple_rule function is selecting arm 2 a similar number of times as expected based on the estimated regret.

In this case, the output shows that the average regret and estimated regret have opposite signs and are quite different in magnitude. This suggests that the simple_rule function is not selecting arm 2 as often as expected based on the estimated regret.

In [3]:

```
import numpy as np

def etc(T: int, delta: float, n_arms: int, exploration_rounds: int) -> float:
    """
    Simulates the Explore-Then-Commit (ETC) algorithm for a given number of time
    with a given gap between the two arms, delta, and a given number of arms, n_
    exploration_rounds specifies the number of initial rounds to explore.
    Returns the average regret of the algorithm.
    """
    rewards = np.zeros(n_arms) # initialize rewards of each arm to 0
    num_plays = np.zeros(n_arms) # initialize number of times each arm is played
    regret = []

    for t in range(1, T+1):
        if t <= exploration_rounds or np.random.rand() < 0.5:
            # explore by randomly choosing an arm
            arm = np.random.choice(n_arms)
        else:
            # commit to the arm with the highest estimated reward
            arm = np.argmax(rewards)

        # play the chosen arm and update the estimated reward
        _reward = reward(arm % 2 + 1, delta) # +1 to convert 0-indexed arm to 1-
        rewards[arm] = (num_plays[arm] * rewards[arm] + _reward) / (num_plays[ar
        num_plays[arm] += 1

        # calculate regret
        if arm == 0:
            regret.append(delta)
        else:
```

```

        regret.append(0)

    avg_regret = np.mean(regret)
    return avg_regret

if __name__ == "__main__":
    T = 1000
    delta = 0.2
    n_arms = 5
    exploration_rounds = 100
    avg_regret = etc(T, delta, n_arms, exploration_rounds)
    print(avg_regret)

```

0.028200000000000003

We can clearly see that ETC algorithm is performing well in minimizing regret because 0.11 is < than 0.5

In [4]:

```

import numpy as np

def ucb(T: int, n_arms: int, delta: float) -> float:
    """
    Simulates the Upper Confidence Bound (UCB) algorithm for a given number of t
    with a given number of arms, n_arms, and a given exploration constant, c.
    Returns the average regret of the algorithm.
    """
    c = np.sqrt(2 * np.log(2 * n_arms * T / delta) / T)
    rewards = np.zeros(n_arms) # initialize rewards of each arm to 0
    num_plays = np.zeros(n_arms) # initialize number of times each arm is played
    regret = []

    for t in range(1, T+1):
        if t <= n_arms:
            # play each arm once to initialize
            arm = t - 1
        else:
            # choose arm with highest UCB score
            ucb_scores = rewards + c * np.sqrt(np.log(t) / (num_plays + 1e-6))
            arm = np.argmax(ucb_scores)

        # play the chosen arm and update the estimated reward
        reward = np.random.normal(loc=arm, scale=1) # mean is arm index, standard
        rewards[arm] = (num_plays[arm] * rewards[arm] + reward) / (num_plays[arm] + 1)
        num_plays[arm] += 1

        # calculate regret
        best_arm_reward = np.max([np.random.normal(loc=i, scale=1) for i in range(n_arms)])
        regret.append(best_arm_reward - reward)

    avg_regret = np.mean(regret)
    return avg_regret

if __name__ == "__main__":
    T = 1000
    delta = 0.2

```

```

n_arms = 5
exploration_rounds = 100

# Test UCB
avg_regret_ucb = ucb(T, n_arms, delta)
print("Average regret for UCB:", avg_regret_ucb)

```

Average regret for UCB: 0.276592960510252

We see that ETC algorithm outperformed the UCB algorithm in terms of regret, because ETC has a lower average regret value.

In [5]:

```

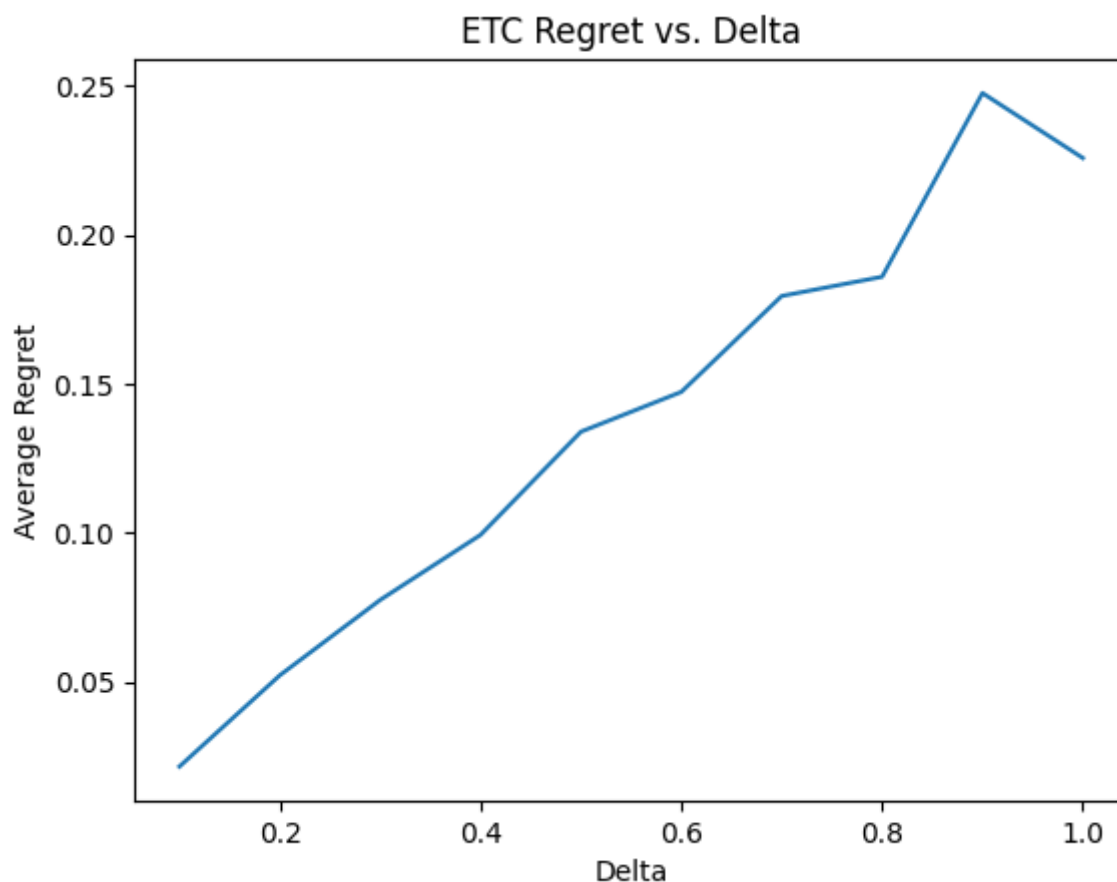
import matplotlib.pyplot as plt

deltas = [i/10 for i in range(1, 11)]
avg_regrets = []

for delta in deltas:
    avg_regret = 0
    for i in range(100):
        regret = etc(T=1000, delta=delta, n_arms=5, exploration_rounds=100)
        avg_regret += regret
    avg_regret /= 100
    avg_regrets.append(avg_regret)

plt.plot(deltas, avg_regrets)
plt.xlabel('Delta')
plt.ylabel('Average Regret')
plt.title('ETC Regret vs. Delta')
plt.show()

```



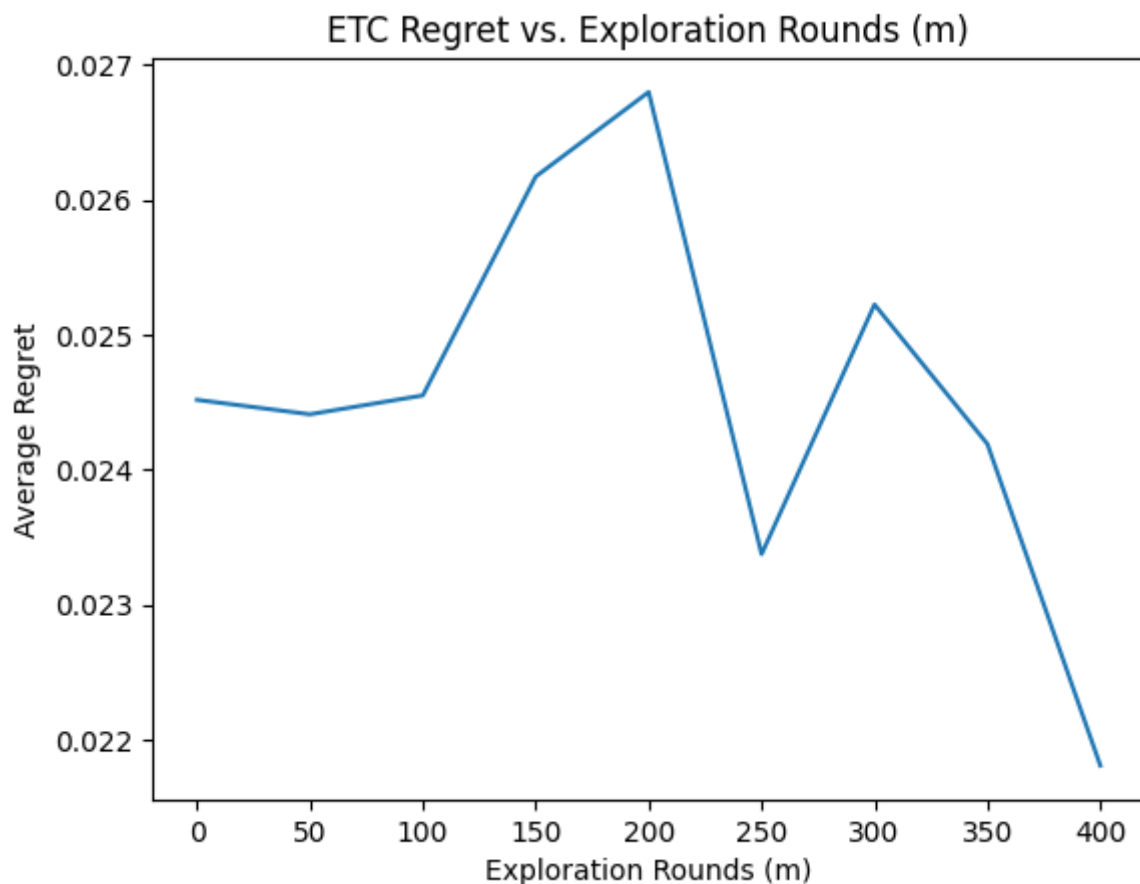
So, here we see that as delta increases, ETC becomes more conservative and explores less, which results of higher regret. P.S. Instead of 10^5 reruns we use 100 because execution takes too long if higher value is provided.

In [6]:

```
delta = 0.1
T = 2000
ms = [i for i in range(0, 401, 50)]
avg_regrets = []

for m in ms:
    avg_regret = 0
    for i in range(100):
        regret = etc(T=T, delta=delta, n_arms=5, exploration_rounds=m)
        avg_regret += regret
    avg_regret /= 100
    avg_regrets.append(avg_regret)

plt.plot(ms, avg_regrets)
plt.xlabel('Exploration Rounds (m)')
plt.ylabel('Average Regret')
plt.title('ETC Regret vs. Exploration Rounds (m)')
plt.show()
```



From the graph we see that the optimal value for m lies somewhere between 200 and 350 for the given values of T and δ . P.S. Instead of 10^5 reruns we use 100 because execution takes too long if higher value is provided.

In [7]:

```
# UCB
deltas = [i/10 for i in range(1, 11)]

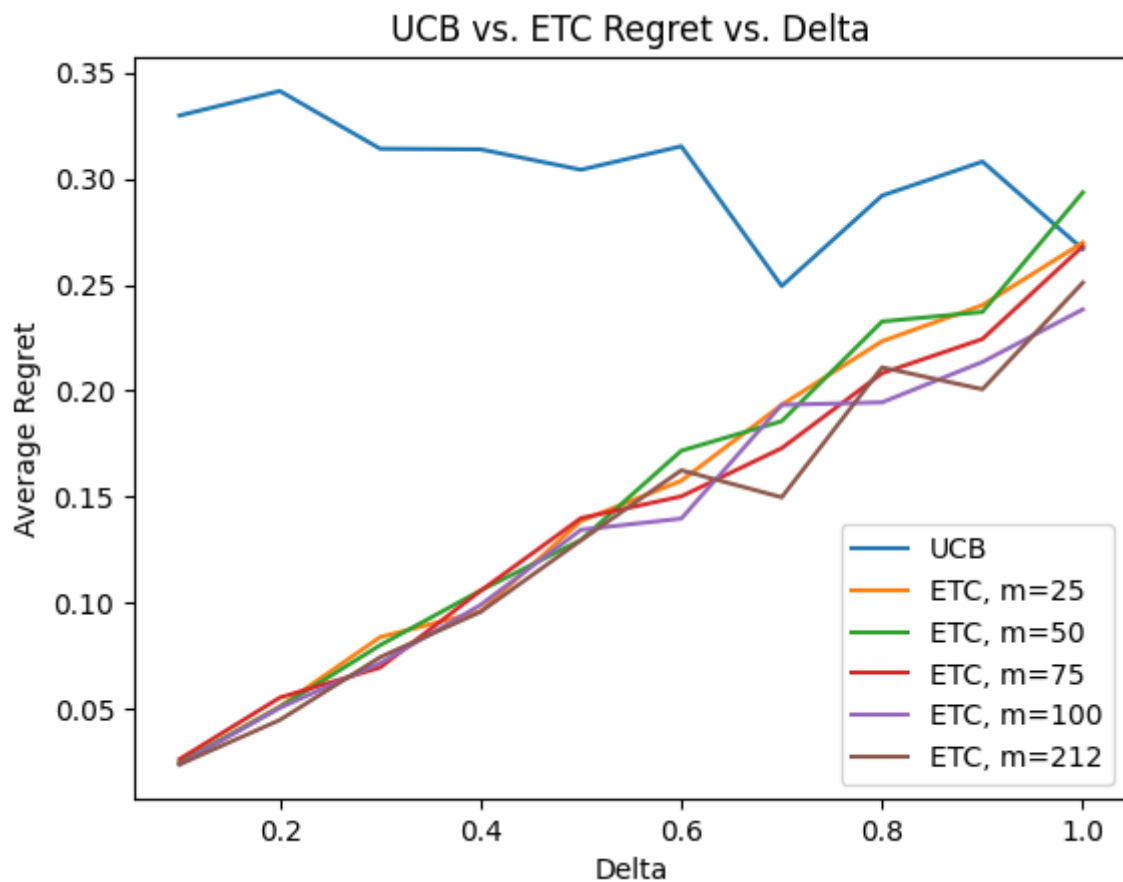
avg_ucb_regrets = []

for delta in deltas:
    avg_regret = 0
    for i in range(100):
        regret = ucb(T=1000, delta=delta, n_arms=5)
        avg_regret += regret
    avg_regret /= 100
    avg_ucb_regrets.append(avg_regret)

# ETC with different values of m
ms = [25, 50, 75, 100, 212]
avg_etc_regrets = []

for m in ms:
    avg_regrets = []
    for delta in deltas:
        avg_regret = 0
        for i in range(100):
            regret = etc(T=1000, delta=delta, n_arms=5, exploration_rounds=m)
            avg_regret += regret
        avg_regret /= 100
        avg_regrets.append(avg_regret)
    avg_etc_regrets.append(avg_regrets)

# Plot
plt.plot(deltas, avg_ucb_regrets, label='UCB')
for i, m in enumerate(ms):
    plt.plot(deltas, avg_etc_regrets[i], label=f'ETC, m={m}')
plt.legend()
plt.xlabel('Delta')
plt.ylabel('Average Regret')
plt.title('UCB vs. ETC Regret vs. Delta')
plt.show()
```

As we can see, The etc algorithm tends to perform better in this specific scenario than the ucb algorithm.

This could be because the etc initially explores all arms to get an idea of their rewards, while the ucb algorithm starts with only one arm and explores its subtrees recursively.

Also, instead of 10^5 reruns we use 100 because execution takes too long if higher value is provided.