

```
In [1]: import math
        from matplotlib import pyplot as plt
        from numpy import random
```

```
In [2]: alpha = 0.1
        Z = 128
        n = 10
        i = [2, 5, 8, 10]
```

```
In [3]: def f_of_B(bits):
        sum = 0
        for i in range(len(bits)):
            sum += pow(2,i) * bits[i]
        return sum
```

```
In [4]: def likelihood_weighting(B):
        exp = abs(Z - f_of_B(B))
        ret = ((1 - alpha) / (1 + alpha)) * pow(alpha, exp)
        return ret
```

```
In [5]: def performSampling(numSamples, bitI):
        numerator = 0.0
        denominator = 0.0

        for i in range(numSamples):
            randomBits = random.randint(2, size = n)
            weight = likelihood_weighting(randomBits)
            denominator += weight
            indicator = randomBits[bitI-1]
            numerator += weight * indicator
        return numerator / denominator
```

```
In [6]: # 3.6b
        # i = {2,5,8,10}
        N = 10
        print("Sampling %d times:" % N)
        bitI = 2
        prob = performSampling(N, bitI)
        print("P(B%d=1|Z=128) = %f" % (bitI, prob))
        bitI = 5
        prob = performSampling(N, bitI)
        print("P(B%d=1|Z=128) = %f" % (bitI, prob))
        bitI = 8
        prob = performSampling(N, bitI)
        print("P(B%d=1|Z=128) = %f" % (bitI, prob))
        bitI = 10
        prob = performSampling(N, bitI)
        print("P(B%d=1|Z=128) = %f" % (bitI, prob))
```

```
Sampling 10 times:
P(B2=1|Z=128) = 1.000000
P(B5=1|Z=128) = 0.000000
P(B8=1|Z=128) = 0.000000
P(B10=1|Z=128) = 0.000000
```

```

In [8]: # 3.6c
# i = {2,5,8,10}
sampleSize = range(100000,1000000,100000)
bit2 = []
bit5 = []
bit8 = []
bit10 = []

for N in sampleSize:
    print("Sampling %d times:" % N)
    bitI = 2
    prob = performSampling(N, bitI)
    bit2.append(prob)
    print("P(B%d=1|Z=128) = %f" % (bitI, prob))
    bitI = 5
    prob = performSampling(N, bitI)
    bit5.append(prob)
    print("P(B%d=1|Z=128) = %f" % (bitI, prob))
    bitI = 8
    prob = performSampling(N, bitI)
    bit8.append(prob)
    print("P(B%d=1|Z=128) = %f" % (bitI, prob))
    bitI = 10
    prob = performSampling(N, bitI)
    bit10.append(prob)
    print("P(B%d=1|Z=128) = %f" % (bitI, prob))

```

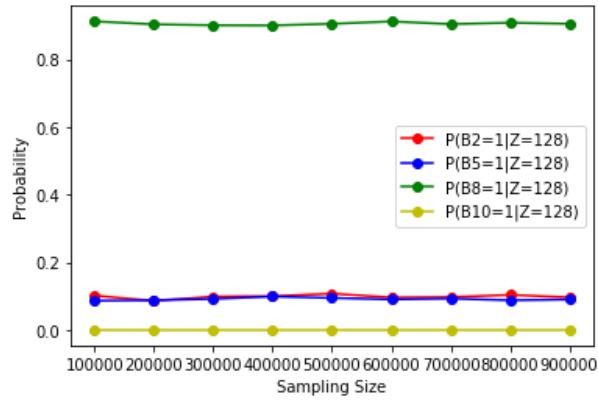
```

Sampling 100000 times:
P(B2=1|Z=128) = 0.101617
P(B5=1|Z=128) = 0.086319
P(B8=1|Z=128) = 0.912466
P(B10=1|Z=128) = 0.000000
Sampling 200000 times:
P(B2=1|Z=128) = 0.085999
P(B5=1|Z=128) = 0.087948
P(B8=1|Z=128) = 0.903759
P(B10=1|Z=128) = 0.000000
Sampling 300000 times:
P(B2=1|Z=128) = 0.098725
P(B5=1|Z=128) = 0.091782
P(B8=1|Z=128) = 0.900945
P(B10=1|Z=128) = 0.000000
Sampling 400000 times:
P(B2=1|Z=128) = 0.099935
P(B5=1|Z=128) = 0.098893
P(B8=1|Z=128) = 0.900463
P(B10=1|Z=128) = 0.000000
Sampling 500000 times:
P(B2=1|Z=128) = 0.107447
P(B5=1|Z=128) = 0.094616
P(B8=1|Z=128) = 0.905299
P(B10=1|Z=128) = 0.000000
Sampling 600000 times:
P(B2=1|Z=128) = 0.096070
P(B5=1|Z=128) = 0.090341
P(B8=1|Z=128) = 0.912219
P(B10=1|Z=128) = 0.000000
Sampling 700000 times:
P(B2=1|Z=128) = 0.097238
P(B5=1|Z=128) = 0.092883
P(B8=1|Z=128) = 0.904366
P(B10=1|Z=128) = 0.000000
Sampling 800000 times:
P(B2=1|Z=128) = 0.103609
P(B5=1|Z=128) = 0.087605
P(B8=1|Z=128) = 0.908406
P(B10=1|Z=128) = 0.000000
Sampling 900000 times:
P(B2=1|Z=128) = 0.096522
P(B5=1|Z=128) = 0.090022
P(B8=1|Z=128) = 0.905336
P(B10=1|Z=128) = 0.000000

```

```
In [9]: plt.plot(sampleSize, bit2, 'o-', color='r', label = 'P(B2=1|Z=128)')
plt.plot(sampleSize, bit5, 'o-', color='b', label = 'P(B5=1|Z=128)')
plt.plot(sampleSize, bit8, 'o-', color='g', label = 'P(B8=1|Z=128)')
plt.plot(sampleSize, bit10, 'o-', color='y', label = 'P(B10=1|Z=128)')

plt.legend()
plt.ylabel('Probability')
plt.xlabel('Sampling Size')
plt.show()
```



```

In [15]: # 3.6c
# i = {2,5,8,10}
sampleSizeExp = range(2,7)
sampleSize = []
bit2 = []
bit5 = []
bit8 = []
bit10 = []

for exp in sampleSizeExp:
    N = pow(10,exp)
    sampleSize.append(N)
    print("Sampling %d times:" % N)
    bitI = 2
    prob = performSampling(N, bitI)
    bit2.append(prob)
    print("P(B%d=1|Z=128) = %f" % (bitI, prob))
    bitI = 5
    prob = performSampling(N, bitI)
    bit5.append(prob)
    print("P(B%d=1|Z=128) = %f" % (bitI, prob))
    bitI = 8
    prob = performSampling(N, bitI)
    bit8.append(prob)
    print("P(B%d=1|Z=128) = %f" % (bitI, prob))
    bitI = 10
    prob = performSampling(N, bitI)
    bit10.append(prob)
    print("P(B%d=1|Z=128) = %f" % (bitI, prob))

```

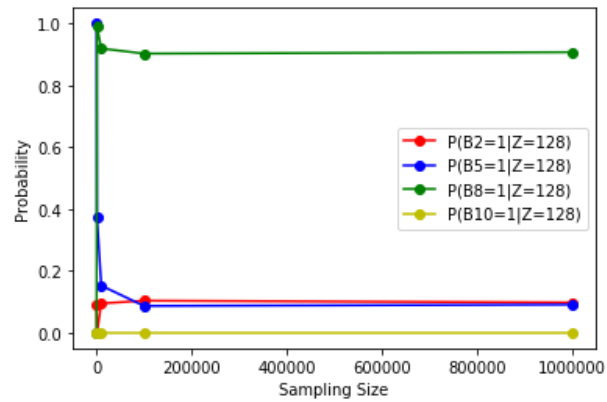
```

Sampling 100 times:
P(B2=1|Z=128) = 0.090910
P(B5=1|Z=128) = 1.000000
P(B8=1|Z=128) = 0.000000
P(B10=1|Z=128) = 0.000000
Sampling 1000 times:
P(B2=1|Z=128) = 0.000005
P(B5=1|Z=128) = 0.372673
P(B8=1|Z=128) = 0.990163
P(B10=1|Z=128) = 0.000000
Sampling 10000 times:
P(B2=1|Z=128) = 0.094771
P(B5=1|Z=128) = 0.152389
P(B8=1|Z=128) = 0.918962
P(B10=1|Z=128) = 0.000000
Sampling 100000 times:
P(B2=1|Z=128) = 0.103722
P(B5=1|Z=128) = 0.086613
P(B8=1|Z=128) = 0.902473
P(B10=1|Z=128) = 0.000000
Sampling 1000000 times:
P(B2=1|Z=128) = 0.097772
P(B5=1|Z=128) = 0.091142
P(B8=1|Z=128) = 0.907009
P(B10=1|Z=128) = 0.000000

```

```
In [16]: plt.plot(sampleSize, bit2, 'o-', color='r', label = 'P(B2=1|Z=128)')
plt.plot(sampleSize, bit5, 'o-', color='b', label = 'P(B5=1|Z=128)')
plt.plot(sampleSize, bit8, 'o-', color='g', label = 'P(B8=1|Z=128)')
plt.plot(sampleSize, bit10, 'o-', color='y', label = 'P(B10=1|Z=128)')

plt.legend()
plt.ylabel('Probability')
plt.xlabel('Sampling Size')
plt.show()
```



In []: