# Solutions to Homework Assignment 2
# CS 6104: Algorithmic Number Theory

**Problem 1.** **[Solution Courtesy of Yizhong Wang]** Chapter 4, Problem 14. You may assume the result in Problem 2.22. Use *Mathematica* to calculate the actual probability for

$$n \in \{100, 200, 300, 400, 500, 600, 700, 800, 900, 1000\}.$$

Put those results in a table that also gives the relative error if the probability is taken to be $6/\pi^2$.

---

We only need to show that

$$\frac{1}{n^2} \sum_{1 \le i \le n} \sum_{1 \le j \le n} i \perp j = \frac{6}{\pi^2} + O(\frac{\log n}{n}).$$

where $i \perp j$ is defined to be 1 if $\gcd(i, j) = 1$, and 0 otherwise.
But,

$$\sum_{1 \le i \le n} \sum_{1 \le j \le n} i \perp j = \sum_{1 \le i \le n} \sum_{1 \le j \le i} i \perp j + \sum_{1 \le i \le n} \sum_{i < j \le n} i \perp j = \sum_{1 \le i \le n} \sum_{1 \le j \le i} i \perp j + \sum_{1 < j \le n} \sum_{1 \le i < j} i \perp j$$

Clearly

$$\phi(n) = \sum_{\substack{1 \le k \le n \\ \gcd(k,n)=1}} 1 = \sum_{1 \le k \le n} k \perp n$$

and for $n \ge 2$,

$$\phi(n) = \sum_{1 \le k < n} k \perp n,$$

hence,

$$\sum_{1 \le i \le n} \sum_{1 \le j \le n} i \perp j = \sum_{1 \le i \le n} \phi(i) + \sum_{1 < j \le n} \phi(j) = 2 * \sum_{1 \le i \le n} \phi(i) - 1$$

From exercise 2.22, we know that

$$\sum_{1 \le k \le n} \phi(k) = \frac{3}{\pi^2} n^2 + O(n \log n)$$

So,

$$\sum_{1 \le i \le n} \sum_{1 \le j \le n} i \perp j = \frac{6}{\pi^2} n^2 + O(n \log n)$$

This leads to

$$\frac{1}{n^2} \sum_{1 \le i \le n} \sum_{1 \le j \le n} i \perp j = \frac{6}{\pi^2} + O(\frac{\log n}{n}).$$

Use the following *Mathematica* code,

```
y={0,0,0,0,0,0,0,0,0,0};
Do[Do[If[GCD[i,j]==1,y[[k]]++],{i,1,k*100},{j,1,k*100}],{k,1,10}];
Do[y[[i]]=y[[i]]/((i*100)^2),{i,1,10}];
k=10;e=6/(Pi^2);
TableForm[Table[{i*100,N[y[[i]],4],100*N[Abs[y[[i]]-e]/e]},{i,1,k}]]
```

get the table:

| **Probability taken as** $6/\pi^2$ | | |
|---|---|---|
| n value | Probability | Relative error(%) |
| 100 | 0.6087 | 0.1271 |
| 200 | 0.6116 | 0.6000 |
| 300 | 0.6088 | 0.1491 |
| 400 | 0.6085 | 0.0891 |
| 500 | 0.6089 | 0.1640 |
| 600 | 0.6083 | 0.0664 |
| 700 | 0.6082 | 0.0506 |
| 800 | 0.6086 | 0.1094 |
| 900 | 0.6082 | 0.0467 |
| 1000 | 0.6084 | 0.0750 |

---

**Problem 2. [Solution Courtesy of Lynn Jones]** Implement the Extended Euclidean algorithm in *Mathematica*. (*Mathematica* has the Euclidean and Extended Euclidean algorithms built in, but do not use those in your implementation.)

Let

$$c_1 = 1717424330343597918506415$$
$$c_2 = 1514122986729833684874188480781$$
$$c_3 = 244480356564695980335975744122413.$$

Show how your implementation can be used to find a solution $x, y, z \in \mathbb{Z}$ to the equation

$$c_1x + c_2y + c_3z = \gcd(c_1, c_2, c_3).$$

Give the *Mathematica* steps to obtain one such solution.

---

Here is my *Mathematica* function definition for the Extended Euclidean algorithm as listed on page 72 of Bach & Shallit:

```
extEuclid[u_,v_] := {                              \
uVar = u;                                          \
varV = v;                                          \
```

```
matr = {{1,0},{0,1}};                              \
n = 0;                                             \
q = 0;                                             \
While[((uVar  != varV  ) && (varV  != 0)), {       \
    q = Floor[uVar /varV ];                        \
    matr = Dot[matr,{{q,1},{1,0}}];                \
    tempU = uVar ;                                 \
    uVar =varV ;                                   \
    varV  = tempU-(q*varV);                        \
    n++;                                           \
}];                                                \
divisor = uVar ;                                   \
ufactor = (-1)^n*matr[[2,2]];                      \
vFact = (-1)^(n+1)*matr[[1,2]];                    \
answers = {divisor, ufactor, vFact} }
```

The function returns a list whose first (and only) member is a list whose members are the divisor $d$ and multipliers $a$ and $b$, such that $d$ is the greatest common divisor of inputs $u$ and $v$ and $au+bv = d$. The only modifications to the algorithm that were made for the *Mathematica* implementation are the addition of the test for $v == 0$ in the While loop condition, and the creation of variable copies for the inputs (*Mathematica* treats the inputs as constants and won't assign to their identifiers).

We can use results of this function to solve the given equation. Let's examine the right side of the equation:

$$\begin{aligned}
\gcd(c_1, c_2, c_3) &= \gcd(c_1, (a_1c_2 + b_1c_3)) \\
&= a_2c_1 + b_2(a_1c_2 + b_1c_3) \\
&= a_2c_1 + b_2a_1c_2 + b_2b_1c_3
\end{aligned}$$

Setting the coefficients of each $c$ equal to the other, we solve for

$$\begin{aligned}
x &= a_2 \\
y &= b_2a_1 \\
z &= b_2b_1
\end{aligned}$$

Here is one way to find a solution with *Mathematica*:

- Set values for variables, $c_1$, $c_2$, and $c_3$.

```
In[5]:= cOne = 171742433034359791850 6415;                  \
        cTwo = 151412298672983368487418848 0781;       \
        cThree = 244480356564695980335975744122413;
```

- Call `extEuclid`$(c_2, c_3)$ and name the output `resultOne`.

```
In[6]:= resultOne = extEuclid[cTwo,cThree]
Out[6]= {{19429427, -422479177174735682644 9601, 2616510555545167714841 6}}
```

- Extract the greatest common divisor, $d$, from `resultOne` and call `extEuclid`$(c_1, d)$, naming the output `resultTwo` (You could also pass in $(a_1 c_2 + b_1 c_3)$, but this is the same result!).

```
In[7]:= resultTwo = extEuclid[cOne,resultOne[[1,1]]]
Out[7]= {{19429427, 0, 1}}
```

- Set $x$, $y$, and $z$ as indicated above.

```
In[8]:= x = resultTwo[[1,2]];                          \
        y = resultTwo[[1,3]] * resultOne[[1,2]];  \
        z = resultTwo[[1,3]] * resultOne[[1,3]];  \
{x,y,z}
Out[8]= {0, -42247917717473568264496O1, 26165105554516771484l6}
```

- Verify the answer.

```
In[9]:= cOne x + cTwo y + cThree z
Out[9]= 19429427
```

---