Math 116 HW8 Jun Ryu

2) a)

```
In [1]: v1 = vector([4,13])
        v2 = vector([-57,-45])
        M = Matrix([v1,v2])
        det(M)
Out[1]: 561

In [2]: (det(M)/(v1.norm()*v2.norm()))^0.5
Out[2]: 0.753621824936351

In [3]: w1 = vector([25453,9091])
        w2 = vector([-16096,-5749])
        (det(M)/(w1.norm()*w2.norm()))^0.5
Out[3]: 0.00110199922846350
```

So, we have determinant of Alice's lattice is $\boxed{561}$.

Hadamard ratio for private basis : $\boxed{0.7536}$

Hadamard ratio for public basis : $\boxed{0.0011}$

b)

```
In [4]: y = vector([155340,55483])
        M.solve_left(y)
Out[4]: (-115993/17, -163408/51)
```

So, we have $e \approx -6823.12 v_1 - 3204.08 v_2$

$\Rightarrow v = \lfloor t_1 \rceil v_1 + \lfloor t_2 \rceil v_2 \Rightarrow v = -6823 v_1 - 3204 v_2$

$\Rightarrow v = (155336, 55481)$

```
In [4]: x = vector([155336,55481])
        N = Matrix([w1,w2])
        N.solve_left(x)

Out[4]: (8, 3)
```

$\Rightarrow v = 8w_1 + 3w_2 \Rightarrow$ plaintext is $m = \boxed{(8,3)}$

Now, we have

$$e = x_1 w_1 + \cdots + x_n w_n + r$$

$$r = e - v = (155340, 55483) - (155336, 55481)$$
$$= \boxed{(4,2)}$$

c) We repeat the above process but using $\{w_1, w_2\}$ instead of $\{v_1, v_2\}$

```
In [8]: N.solve_left(y)

Out[8]: (-428/51, -1169/51)
```

So, $e \approx -8.39 w_1 - 22.92 w_2$

$\Rightarrow v = -8w_1 - 23w_2 \Rightarrow$ resulting "plaintext" is $(-8,-23)$, which is incorrect.

3) We have $v_1 = (1,1)$ and $v_2 = (2, 0.5)$

We have $\|v_1\|^2 = 1^2 + 1^2 = 2$ and

$\|v_2\|^2 = 2^2 + 0.5^2 = 4.25$ so

$\|v_1\| < \|v_2\|$ ✓

Now, we compute $m = \left\lfloor \dfrac{(1,1)\cdot(2,0.5)}{2} \right\rceil = \lfloor 1.25 \rceil = 1$

So, we have $v_2 = (2, 0.5) - 1\cdot(1,1) = (1, -0.5)$

Now, we repeat the process:

$\|v_1\|^2 = 2$, $\|v_2\|^2 = 1^2 + (-0.5)^2 = 1.25$

So we swap v_1 and v_2. $(v_1 = (1,-0.5),\ v_2 = (1,1))$

Now, we compute

$$m = \left\lfloor \frac{(1,-0.5)\cdot(1,1)}{1.25} \right\rceil = \lfloor 0.4 \rceil = 0$$

So, since $m = 0$, we have $\boxed{w_1 = (1,-0.5),\ w_2 = (1,1)}$

```
In [6]: v1 = vector([1,1])
        v2 = vector([2,0.5])
        M = Matrix([v1,v2])
        (det(M)/(v1.norm()*v2.norm()))^0.5

Out[6]: 0.603161203621801*sqrt(-sqrt(2))
```

← Hadamard ratio for $\{v_1, v_2\}$

```
In [7]: w1 = vector([1,-0.5])
        w2 = vector([1,1])
        (det(M)/(w1.norm()*w2.norm()))^0.5

Out[7]: 0.819036258812720*sqrt(-sqrt(2))
```

← Hadamard ratio for $\{w_1, w_2\}$

4)

```
In [1]: def lattice_reduction(v1,v2):
            rounded_m = 1
            counter = 0
            while rounded_m != 0:
                if v2.norm() < v1.norm():
                    v2_copy = v2
                    v2 = v1
                    v1 = v2_copy
                m = (v1.dot_product(v2))/(v1.norm()^2)
                rounded_m = round(m)
                v2 = v2 - rounded_m*v1
                counter = counter + 1
            return (v1,v2,counter)

In [2]: v1 = vector([120670,110521])
        v2 = vector([323572,296358])
        lattice_reduction(v1,v2)

Out[2]: ((14, -47), (-362, -131), 7)

In [3]: M = Matrix([v1,v2])
        (det(M)/(v1.norm()*v2.norm()))^0.5

Out[3]: 0.000512357132355632

In [4]: w1 = vector([14,-47])
        w2 = vector([-362,-131])
        (det(M)/(w1.norm()*w2.norm()))^0.5

Out[4]: 0.999167162226088
```

So, we have our solution to the SVP is $v_1 = \boxed{(14, -47)}$. This algorithm took $\boxed{7}$ steps.

The Hadamard ratio of the input was $\boxed{0.0005}$

The Hadamard ratio of the output was $\boxed{0.9992}$