2. wort-case time of Karatsuba algorithm.

Step 1. $T_w(n) = n(3 \cdot T_w(\frac{n}{2})) + a \cdot n$.

Step 2. Guess. $T_w(n) = O(n^x) \leq C \cdot n^x$. $c > 0$.

Step 3. $T_w(n) = 3n \cdot T_w(\frac{n^2}{2}) + a \cdot n$

$\leq 3 \cdot c \cdot (\frac{n}{2})^x + a \cdot n$

$\leq C \cdot n^x$.

$3c(\frac{n}{2})^x + a \cdot n \leq C \cdot n^x$

$3c\frac{n^x}{2^x} + a \cdot n \leq Cn^x$

$\frac{3}{2^x} + \frac{an}{Cn^x} \leq 1$

send $n \to \infty$:

$\frac{3}{2^x} \leq 1$.

$2^x > 3$

$x > \log_2 3 \approx 1.59$.

So, Karatsuba runstime wrst-case $O(n^{1.59})$

3. $T(n)$ the to multiply two bit string each of length n.

$$T(n) \cdot 2T\left(\frac{n}{2}\right)$$

$$= (n)\left(\frac{n}{2}\right)$$

$$= \frac{n^2}{2} \text{ bit length}$$

Then we have two n string. $= O(n^{1.59})$

$$= O\left(\left(\frac{n^2}{2}\right)\log\frac{3}{2}\right)$$

$$= O(n^3)$$

$$T(n) \cdot 2T\left(\frac{n}{2}\right) + O(n^3)$$

So the worst-case complexity of better Karatsuba is $O(n^3)$

4. 1. algorithm (cube, number Air)

2. (cube $\cdot \frac{1}{2}$, number Air) half the cube and run algorithm.

3. if (cube has 2 Airplanes, check if has minimum dist)

else.

in two cube check boundary of first cube near to boundary of second cube.

4. repeat until get minimum.