4.2-3

To use Strassen algorithm on matrix with n as not exact power of 2, we can extend the matrix to nearest $n' * n'$ where n' is power of 2

For example, we have $6 * 6$ matrix mutiplication we can extend it to $8 * 8$ by adding zeroes.

Since n cannot be extended twice or more, we have $n \leq 8 \Rightarrow T(n) = \Theta(n^{\lg 7})$

4.2-4

split $n \times n$ matrix into $n/3 * n/3$ matrices. with k multiplications.

Running time will be $T(n) = kT(n/3) + \Theta(n^2)$

Now, we need to compare

$n^{\lg_3 k}$ and n^2

since we are trying to find the largest k, with $O(n^{\lg 7})$ running time

$k \leq 3^{\lg 7} \approx 21.9$

so the largest k is 21

$\lg_3 k > 2$ $\quad T(n) = \Theta(n^{\lg_3 k})$

$= \Theta(n^{\lg_3 21})$

$= \Theta(n^{2.8})$

4.2-5

68 * 68 :

$$T_{68}(n) = 132464\, T\left(\frac{n}{68}\right) + n^2$$
$$= \Theta\left(n^{\log_{68} 132464}\right) \approx \Theta\left(n^{2.795128}\right)$$

70 * 70 :

$$T_{70}(n) = 143640\, T\left(\frac{n}{70}\right) + n^2$$
$$= \Theta\left(n^{\log_{70} 143640}\right) \approx \Theta\left(n^{2.795128}\right)$$

72 * 72 :

$$T_{72}(n) = 155424\, T\left(\frac{n}{72}\right) + n^2$$
$$= \Theta\left(n^{\log_{72} 155424}\right) \approx \Theta\left(n^{2.795147}\right)$$

$$T_{70} < T_{68} < T_{72}$$

Therefore, asymptotically, 70 * 70 is fastest and faster than strassen's algorithm.

4.2-7

$(a+bi)\cdot(c+di) = (ac-bd) + (ad+bc)i$

We need to use 3 multiplizations to calculate both the real and complex parts.

Let $P_1 = (a+b)\cdot(c+d)$

$= ac + bc + ad + bd$

$P_2 = a\cdot c$ $\quad$ $P_3 = b\cdot d$

real part : $(ac-bd) = P_2 - P_3$

Complex part : $(ad+bc) = P_1 - P_2 - P_3$

Hence, we have the solution.

4.5-1

a) $T(n) = 2T\left(\frac{n}{4}\right) + 1$

$\log_b a = \log_4 2 = 0.5$

$n^{\log_b a} = n^{0.5}$ and $f(n) = 1 = n^0$

$n^{\log_b a} > f(n)$

therefore $T(n) = \Theta\left(n^{\log_b a}\right) = \Theta(n^{0.5})$

b) $T(n) = 2T\left(\frac{n}{4}\right) + \sqrt{n}$

$\log_b a$ is the same as the last one $\Rightarrow 0.5$

$f(n) = \sqrt{n} = n^{0.5}$

$n^{\log_b a} = f(n)$

Therefore

$T(n) = \Theta\left(n^{\log_b a} \lg n\right) = \Theta(\sqrt{n} \lg n)$

c) $T(n) = 2T\left(\frac{n}{4}\right) + n$

$\log_b a = 0.5 \qquad f(n) = n^1$

$f(n) > \log_b a$ Therefore

$T(n) = \Theta\left(f(n)\right) = \Theta(n)$

d) $T(n) = 2T\left(\frac{n}{4}\right) + n^2$

$\log_b a = 0.5 \qquad f(n) = n^2$

$f(n) > \log_b a$ Therefore

$T(n) = \Theta\left(f(n)\right) = \Theta(n^2)$

4.5.2

Asymptotical running time of strassen's algorithm is $\Theta(n^{\lg 7})$

$$T(n) = aT\left(\frac{n}{4}\right) + n^2$$

let's assume $\log_4 a > 2$, that is $a > 16$

$\log_4 a > f(n)$ so first case of master theorem.

$$T(n) = \Theta(n^{\log_4 a})$$

$$\Theta(n^{\log_4 a}) < \Theta(n^{\lg 7})$$

$\log_4 a < \lg 7$ $\sqrt{a} < 7$

$\lg \sqrt{a} < \lg 7$ $a < 49$

So, largest a is 48

4.5-3

$$T(n) = T(\frac{n}{2}) + 1$$

$$\log_b a = \log_2 1 = 0 \quad f(n) = 1$$

$$\log_b a < f(n)$$

So, It is case 2 of master theorem

$$T(n) = \Theta(n^{\log_2 1} \lg n)$$

$$= \Theta(1 \cdot \lg n)$$

$$= \Theta(\lg n)$$

4.5-5

case 3 is that $n^{\log_b a} < f(n)$

let $a = 8$ $b = 2$

$f(n) = n^2$

$n^{\log_b a} = n^3 > f(n) = n^2$

9.3-2

To make it $O(n \lg n)$ worst case, we can modify the select process. Instead of partition on a fixed index, we can use Select to find the ith smallest item in a given fixed sized array.

9.3-5

lets say we want to find ith order statistics,

we can use median to find the position for partition.

Then check whethere median position +1 is i, if so, return median position +1

if i < median position +1, we do the process on the left half of array

eles, do the process of right array.

repeat till it returns result.

$$T(n) = T(n/2) + O(n) = O(n)$$

30.a

$$(ax+b)\cdot(cx+d) = acx^2 + \underbrace{adx + bcx}_{=(ad+bc)x} + bd$$

$$P_1 = (a+b)(c+d) = ac + cb + ad + cd$$

$$P_2 = a\cdot c \qquad P_3 = c\cdot d$$

$$(ad+bc) = P_1 - P_2 - P_3$$

b: Low. high half:

$$H_i(x) = \sum_{j=n/2}^{n-1} a_{ij} x^j \qquad L_i(x) = \sum_{j=0}^{n/2-1} a_{ij} x^j$$

$$P_1(x) = H_1(x)\cdot x^{n/2} + L_1(x)$$

$$P_2(x) = H_2(x)\cdot x^{n/2} + L_2(x)$$

$$P_1(x) \cdot P_2(x) = x^n H_1(x) H_2(x) + H_1(x) \cdot L_2(x) \cdot x^{n/2}$$
$$+ L_1(x) \cdot H_2(x) x^{n/2} + L_1(x) \cdot L_2(x)$$

This yield same result as part a.

$$T(n) = 3T(n/2) + \theta(n) = \theta(n^{\lg 3})$$

odd and even:

$$O_i(x) = \sum_{j=0}^{n/2-1} a_{i,2j+1} x^j$$

$$E_i(x) = \sum_{j=0}^{n/2-1} a_{i,2j} x^j$$

$$P_1(x) = x O_1(x^2) + E_1(x^2)$$

$$P_2(x) = x O_2(x^2) + E_2(x^2)$$

$$P_1(x) P_2(x) = x^2 (O_1(x^2) \cdot O_2(x^2)) + x(O_1(x^2) \cdot E_2(x^2) + O_2(x^2) \cdot E_1(x^2))$$
$$+ E_1(x^2) \cdot E_2(x^2)$$

$$T(n) = 3T(n/2) + \theta(n) = \theta(n^{\lg 3})$$

C:

$$P_1 = \sum_{k=0}^{\lfloor \lg A_1 \rfloor} a_{k,1} 2^k$$

$$P_2 = \sum_{k=0}^{\lfloor \lg A_2 \rfloor} a_{k,2} 2^k$$