① pseudo code:

```
SelectionSortR(arr, n):
base case O(1)            [ if n == len(arr):
                          [     return
worst case O(n)           [ K = getminIndex(arr, n, len(arr)-1)
Swapping O(1)             [ if K != n
                          [     swap(arr[K], arr[n])
recursive case T(n-1)     [ SelectionSortR(arr, n+1)
```

so, $T(n) = \begin{cases} T(n-1) + O(n) & , \text{if } n > 1 \\ O(1) & , \text{if } n \le 1 \end{cases}$

② a) False, counterexample:

$f(n) = \log_2 n \qquad g(n) = \log_8 n$

$$\lim_{n\to\infty} \frac{\log_2 n}{\log_8 n} = 3$$

$$\lim_{n\to\infty} \neq 0 \Rightarrow \Omega \quad , \neq \infty \not\Rightarrow \omega$$

$\square$

b) True, proof: (note merge sort is $n\log n$)

$f(n) = n\log n \qquad g(n) = n^2$

$$\lim_{n\to\infty} \frac{n\log n}{n^2} = \lim_{n\to\infty} \frac{\log n}{n} = 0$$

so, $O(n\log n) = O(n^2)$ $\square$

③ Proof:

($\Longrightarrow$) Assume $f = O(g)$ and $f = \Omega(g)$ to prove that $f = \Theta(g)$.

Then $\exists\, a > 0 \ni a\,g(n) \le |f(n)| \;\forall n > n_1$ (by definition Ω)

And $\exists\, b > 0 \ni |f(n)| \le b\,g(n) \;\forall n > n_0$ (by definition O).

Hence, $\exists\, a, b > 0 \ni a g(n) \le |f(n)| \le b(g)$ $\forall n > \max(n_0, n_1) \Longrightarrow f = \Theta(g)$.

($\Longleftarrow$) Assume $f = \Theta(g)$ to prove $f = O(g)$ and $f = \Omega(g)$.

Then $\exists\, c_1, c_2 > 0 \ni c_1 g(n) \le |f(n)| \le c_2 g(n)$ $\forall n > n_0$, (by definition Θ).

So, $\forall n > n_0 \;\exists\, c_1 > 0 \ni c_1 g(n) \le |f(n)|$ and, $\forall n > n_0 \;\exists\, c_2 > 0 \ni |f(n)| \le c_2 g(n)$ $\Longrightarrow f = O(g)$ and $f = \Omega(g)$.

$\square$

(4)

	f	g	O	o	Θ	$\sim$	Ω	ω
(1)	n^2	$9n^2$	✓		✓		✓	
(2)	n^3+4	$n^3+\sin t \log(n)$	✓		✓	✓	✓	
(3)	2^n	n					✓	✓
(4)	$\log(2n)$	n	✓	✓				
(5)	$n\log(n)$	n^n	✓	✓				

(5) <u>note</u> : properties of logarithms

$$\log(b)^a = a \cdot \log(b)$$

So, $\log_2(n)^{\log_2 17} = \log_2 17 \cdot \log_2 n$

and, $\log_2(17)^{\log_2 n} = \log_2 n \cdot \log_2 17$

They are the same function!!!

But, ... we can prove Θ anyways

$$\lim_{n\to\infty} \frac{\log_2 17 \cdot \log_2 n}{\log_2 n \cdot \log_2 17} = \lim_{n\to\infty} 1 = 1$$

$\square$

(Hence they are also $\sim$).

(b) Strassen's recurrence:

$$T(n) = \begin{cases} \Theta(1) & \text{, if } n=1 \\ 7T(n/2) + \Theta(n^2) & \text{, if } n>1 \end{cases}$$

By master theorem:

$a=7, b=2, f(n) = \Theta(n^2)$

$\Rightarrow n^{\log_b a} = n^{\log_2 7}$

Case 1: Let $\varepsilon = \log_2 \frac{7}{4} > 0$, then

$f(n) = \Theta(n^2) = O(n^2) = O(n^{\log_2 7 - \log_2 \frac{7}{4}})$

$= O(n^2)$

Thus, $T(n) = \Theta(n^{\log_2 7})$ $\square$