1

a) $\lim_{n \to \infty} \dfrac{n^2+7n}{n^3+7} = 0 \therefore f(n) = O(n^3+7)$

b) $\lim_{n \to \infty} \dfrac{12n + \log_2 n^2}{n^2+6n} = 0 \therefore f(n) = O(n^2+6n)$

c) $\lim_{n \to \infty} \dfrac{n \log_2(3n)}{n + \log_2(8n^3)} = \infty \therefore f(n) = \Omega(n + \log_2(8n^3))$

d) $\lim_{n \to \infty} \dfrac{n^n+5n}{3 \cdot 2^n} = \infty \therefore f(n) = \Omega(3 \cdot 2^n)$

e) $\lim_{n \to \infty} \dfrac{\sqrt[3]{2n}}{\sqrt{3n}} = 0 \therefore f(n) = O(\sqrt{3n})$

2

a) $f(n) = n+1$

let $g(n) = n$

$$\lim_{n\to\infty} \frac{n+1}{n} = 1 \therefore f(n) = O(n)$$

b) $f(n) = 3(n+1) + 1$

$f(n) = 3n + 4$

let $g(n) = n$

$$\lim_{n\to\infty} \frac{3n+4}{n} = 3 \quad \therefore f(n) = O(n)$$

c) $f(n) = n+1$

let $g(n) = n$

$$\lim_{n\to\infty} \frac{n+1}{n} = 1 \quad \therefore f(n) = O(n)$$

d) $f(n) = 5 \quad \therefore f(n) = O(1)$

(3)

a) $f(n) = n = O(n)$ (Without loop)

b) $f(n) = n+1 = O(n)$ (with loop)

Hence ~~both~~ functions' complexities are $O(n)$ writing with loop can be adventageous of writing if repetion is ~~too~~ many.

(4) Pseudo-code

```
for i=0; i<n; i++:
        if arr[i] == target number
                return true
return false
```

$f(n) = n+1$, let $g(n) = n$

$\lim_{n \to \infty} \frac{n+1}{n} = 1$, $f(n) = \Omega(n) = \Theta(n) = O(n)$

lower bound of $f(n)$ is $\Omega(n)$. Therefore program cannot run at constant ~~time~~.

(5)

```
for i=0, minA = A[0]; i<n; i++ :
    if A[i] < minA:
        minA = A[i]
for i=0, minB = B[0]; i<m; i++ :
    if B[i] < minB:
        minB = B[i]
return (minA * minB)
```

n+1, O(n)

m+1, O(m)

$f(n) = n + m + 2$

$f(n) = O(\max(n, m))$