1.

(a) $6n^2 - 41n + 2 \in O(n^2)$

$$6n^2 - 41n + 2 \le 6n^2 + 2n^2$$
$$= 8n^2$$
$$= 8\, g(n)$$

$\therefore$ There exist $c = 8$ and $n_0 = 1$ such that $\forall\, n \ge n_0 : 6n^2 - 41n + 2 \in cn^2$

$\therefore\ 6n^2 - 41n + 2 \in O(n^2)$

(b) $\forall\, a \ge 1 : 2^n \in O(2^{n-a})$

Let $C = 2^a$ and $n_0 = 1$ then for $\forall\, n \ge n_0$ and $a \ge 1$,

$$2^n = 2^a \cdot 2^{n-a}$$
$$= c\, 2^{n-a}$$

$\therefore\ \forall\, a \ge 1 : 2^n \in O(2^{n-a})$

(c) $\forall a > 1 : O(\log_2 n) \in O(\log_a n)$

$$\log_2 n = \frac{\log_a n}{\log_a 2}$$

$$\leq \log_a n$$

$$= \log_a 2 \quad g(n)$$

$\therefore$ There exist $c = \log_a 2$ and $n_0 = 1$

such that $\forall n \geq n_0, a > 1 : \log_2 n \leq c \log_a n$

$\therefore \forall a > 1 : O(\log_2 n \in O(\log_a n)$ //

(d) $\forall a > 1 : a^{a^{n+1}} \in O(a^{a^n})$

If $\forall a > 1 : a^{a^{n+1}} \in O(a^{a^n})$

then there exist C and n_0 such that

$$\forall a > 1, n > n_0 \quad a^{a^{n+1}} \leq a^{a^n} \quad \dots (i)$$

left hand side :

take $\log_a$, then

$$a^{n+1}$$

$$= a \cdot a^n \quad \dots (ii)$$

right hand side :

take $\log a$, then

$$a^n \quad \dots (iii)$$

$\because$ (ii), (iii), and $a > 1$

$$a^{a^{n+1}} \geq a^{a^n} \quad \dots (iv)$$

(i) contradict (iv)

$$\therefore \forall a > 1 : a^{a^{n+1}} \notin O(a^{a^n})$$

$\#$

(e) If $f_1(n) \in O(g_1(n))$ and $f_2(n) \in O(g_2(n))$,
then $f_1(n) + f_2(n) \in O(g_1(n) + g_2(n))$

$\because f_1(n) \in O(g_1(n))$

There exist $n_{01}, C_1, \forall n \geq n_{01}: f_1(n) \leq C_1 g_1(n)$ ... (i)

Similarly,

There exist $n_{02}, C_2, \forall n \geq n_{02}: f_2(n) \leq C_2 g_2(n)$ ... (ii)

(i) + (ii)

$$f_1(n) + f_2(n) \leq C(g_1(n) + g_2(n))$$

$\therefore$ If $f_1(n) \in O(g_1(n))$ and $f_2(n) \in O(g_2(n))$,
then $f_1(n) + f_2(n) \in O(g_1(n) + g_2(n))$ //

2

```
(a) for (i = 1; i < n-8; i++) {
        for (j = i; j < i+8; j = j++) {
            <some-constant number of atomic/elementary operations>
    }}
```

$$\sum_{i=1}^{n-8} \sum_{j=i}^{i+8} C$$

$$= \sum_{i=1}^{n-8} C \cdot (i+8-i+1)$$

$$= \sum_{i=1}^{n-8} C \cdot 9$$

$$= 9C\,(n-8)$$

$$\therefore \quad O(n)$$

//

```
(b) for i in the range [1, n] {
        for j in the range [i, n] {
            for k in the range [1, j-i] {
              <some-constant number of atomic/elementary operations>
    }}}
```

$$\sum_{i=1}^{n}\sum_{j=i}^{n}\sum_{k=1}^{j-i} C$$

$$= \sum_{i=1}^{n}\sum_{j=i}^{n} C(j-i)$$

$$= \sum_{i=1}^{n}\left(C\sum_{j=i}^{n} j + Ci\sum_{j=i}^{n}\right)$$

$$= \sum_{i=1}^{n}\left(C\sum_{j=1}^{n} j - C\sum_{j=1}^{i} j + Ci\sum_{j=i}^{n}\right)$$

$$= \sum_{i=1}^{n}\left(C\frac{1}{2}n(n+1) - C\left(\frac{1}{2}i(i+1)\right) + Ci(n-i+1)\right)$$

$$= C\left(\sum_{i=1}^{n}\left(-\frac{3}{2}i^2\right) + \sum_{i=1}^{n}\left(\frac{1}{2}+n\right)i + \sum_{i=1}^{n}\frac{1}{2}n(n+1)\right)$$

$$= C\left(-\frac{3}{2}\cdot\frac{1}{6}n(n+1)(n+2) + \left(\frac{1}{2}+n\right)\cdot\frac{n(n+1)}{2} + \frac{1}{2}n^2(n+1)\right)$$

$$O(n^3)$$

//

(c)
```
x = pow(2, n);
i = 1;
while i <= x {
  for j in range [1, i] {
        <some-constant number of atomic/elementary operations>
  }
  i = i * 2
}
```
Assume that pow(2, n) (i.e., 2^n) is computed *magically* in constant time.

The inside loop takes $O(n)$ because it runs $[1, i]$

The ouside while loop takes $O(\log n)$ because the upper bound, x is constant but i increases exponentially

$\therefore \; O(n \log n)$

Extra credit

```
function myf(integer a, integer n) {
    integer a1;
    while n >= 0 {
        if n==0 then return 1;          → O(1)
        a1 = myf(a, n/2); *             → O(log n)
        if n is even then  {
            return a1 * a1;             → O(1)
        }
        else {
            return a * a1 * a1;         → O(1)
        }
        n = n/2;
    }
}
```

Assume that $n/2 = 0$, when $n < 2$.

* call the function itself recursively with decreasing n by half

$\Rightarrow$ $O(\log n)$

The while loop takes $O(\log n)$ because everytime it gets into the loop, it decrease n by half while the bound stays the same

$\therefore$ Inside of loop is $O(\log n)$ and the loop is $O(\log n)$

$$O((\log(n))^2)$$

/1