a $5n^3 + 2n^2 + 3n = O(n^3)$

$5n^3 + 2n^2 + 3n < 5n^3 + 2n^3 + 3n^3 \quad c_1 \cdot n^3$

$5n^3 + 2n^3 + 3n^3 < c_1 \cdot n^3 \qquad n > 1$

$10n^3 < c_1 \cdot n^3$

$c_1 = 11 \qquad \boxed{n_0,\ n > 1}$

b) $\sqrt{7n^2 + 2n - 8} = \Theta(n)$

$c_1 \cdot n < \sqrt{7n^2 + 2n - 8} < c_2 \cdot n$

$c_1 \cdot n < \sqrt{7n^2 + 2n^2} < c_2 \cdot n$

$c_1 \cdot n < \sqrt{9n^2} < c_2 \cdot n$

$c_1 \cdot n < 3n < c_2 \cdot n \qquad c_1 = 2 \quad n > 1$

$c_2 = 4$

c) $O(f(n)) \cdot O(g(n)) = O(f(n)\, g(n))$

$0 < f(n) < c_1 f(n) \cdot 0 < g(n) < c_2 \cdot g(n) = 0 < f(n)\, g(n) < c_3 \cdot f(n)\, g(n)$

$c_1 f(n) \cdot c_2\, g(n) = c_3\, f(n)\, g(n)$

$c_1 c_2 = c_3 \qquad n > 1$

2) def example 1:

```
n = len list     θ(1)
total = 0        θ(1)
for j in range(n):
    for k in range(j+1)
        total += lst[k]    n-1 work    n work    θ(n^2)
return total  θ(1)
```

example 1 $\theta(n^2)$

def example 2:

```
n = len(lst)    θ(1)
prefix = 0
total = 0
for j in range(n)
    prefix += lst[j]    } constant operation    done n times    θ(n)
    total += prefix
return total
```

example 2, $\boxed{\theta(n)}$

Scanned by CamScanner

c) 1, 2, 4, 8, 16, 32, 64] code stops at 64

to find ~~with~~ n work bit log

ung $\log_2(n) \cdot n = 2\log(n) = \theta(\log_2 n)$

d) def change 4 (n):

i = n, sum = 0

while (i ≥ 1):

for j in range(i)

sum += j·j

i // 2

return sum

$\left[n + \frac{n}{2} + \frac{n}{4} + \frac{n}{8} + \ldots 1 \right]$ calculate how much work is done

$$\sum_{i=0}^{\log_2 n} n = n \cdot \left(\frac{1 - \frac{1}{2}^{\log_2 n}}{1 - \frac{1}{2}} \right)$$

$$n \cdot \frac{\left(1 - 2^{-\log_2 n}\right)}{\frac{1}{2}}$$

$$2n \cdot 1 - \frac{1}{2^{\log_2 n}} \quad \frac{1}{n}$$

$$2n\left(1 - \frac{1}{n}\right) = 2n - 2 = \theta(n)$$

Scanned by CamScanner