1.10

- $f_1(n) = n^2$ and $f_2(n) = n^2 + 1000n$

$f_1(n)$ is $O(f_2(n))$ if $\exists$ $c, n_0 > 0$ such that $f_1(n) \leq f_2(n)$

$n^2 \leq c(n^2 + 1000n)$ for $n_0, c > 0$

$n_0 = 1$ and $c = 1 \Rightarrow$ for $n \geq 1$, $n^2 \leq n^2 + 1000n$

$f_2(n)$ is $\Theta(f_1(n))$ if $\exists$ $c, n_0 > 0$ such that $f_2(n) \leq f_1(n)$

$n^2 + 1000n \leq cn^2$

As $n \to \infty$ $n^2 + 1000n \simeq cn^2$ and $\lim\limits_{n\to\infty} \frac{cn^2}{n^2+1000n} = \lim\limits_{n\to\infty} \frac{cn^2}{n^2} = C$

As $n \to \infty$, $f_2(n)$ is $\Theta(f_1(n)) \Rightarrow f_1$

- $f_1(n) = n^2$ $\qquad f_3(n) = \begin{cases} n & \text{if } n \text{ is odd} \\ n^3 & \text{if } n \text{ is even} \end{cases}$

f_3 is not continous so there is nowhere where they cross each other or one is a upper or lower bound of the other

- $f_1(n)$ and $f_4(n)$

  $f_1(n)$ is $O(f_4(n))$ if $\exists\ c, n_0 > 0$ such that $f_1(n) \leq f_2(n)$

  $n^2 \leq cn^3$

  $n_0 = 101$ and $c = 1 \Rightarrow$ For $n \geq 101$, $n^2 \leq n^3$

  $f_1(n)$ is $O(f_4(n))$ and $f_4(n)$ is $\Omega(f_1(n))$

- $f_2(n)$ and $f_3(n)$

  f_3 is $O(f_2(n))$ if $\exists\ c, n_0 > 0$ such that $f_2(n) \leq cf_3(n)$

  $n^2 + 1000n \leq cn^3$

  $n_0 = 1$ and $c = 1$, For $n \geq 1$ $n^2 + 1000n \leq n^3$

  f_3 is $O(f_2(n))$

- $f_3(n)$ and $f_4(n)$

  f_3 is not continuous so we cannot really tell

- [illegible]

1.12

a- Each loop runs n time and they are all nested so the runtime will be $n \times n \times n \Rightarrow O(n^3)$

$$\boxed{O(n^3)}$$

b. Procedure mystery (n: integer);
  var
    i, j, k: integer
  begin
③   for i := 1 to n - 1 do
②   for j := i+1 to n do
①   for k := 1 to j do
            { some statement requiring O(1) time }
          end
        end
      end

① is $\sum_{k=j}^{j} 1$ ② is $\sum_{j=i+1}^{n} j$ ③ is $\sum_{i=1}^{n-1} n$

The total runtime is

$$\sum_{i=1} \sum_{j=i+1} \sum_{k=j} 1 \Rightarrow \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} J = (n-1)\left(\frac{n^2+n}{2}\right)\left(\sum_{i=1}^{n-1} - \frac{i^2}{2} - \frac{i}{2}\right)$$

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

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

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

$$= \frac{n^3-n}{3} \Rightarrow \boxed{O(n^3)}$$

c. The first for loop and the function odd(i) will run $n/2$ times

The second for loop runs for $(n-i+1)$

The third for loop runs for $(i-1+1)$

The total runtime is $\frac{n}{2}(n-i+1+i-1+1) = \frac{n}{2}(n+1) = \frac{n^2}{2}+n$

The function will be $\boxed{O(n^2)}$

d.

$T(n) = 2T(n-1) + O(1)$

$T(n-1) = 2T(n-2) + O(1)$

$T(n-2) = 2T(n-3) + O(1)$

$T(n-1) = 2(2T(n-3)+O(1)) + O(1)$

$T(n) = \underbrace{2\times2((2}_{2^i}\,T(\underbrace{n-3}_{n-i}) + O(1)) + O(1)) + O(1)$

$= 2^i\,T(n-i) + O(1) \Rightarrow \boxed{O(2^n)}$

$\vdots$