Task 1

A	B	O	o	Ω	ω	Θ
$\lg^k n$	n^{ε}	+	+	−	−	−
n^k	c^n	+	+	−	−	−
$\sqrt{n}$	$n^{\sin n}$	−	−	−	−	−
2^n	$2^{n/2}$	−	−	+	+	−
$n^{\log c}$	$c^{\log n}$	+	−	+	−	+
$\log(n!)$	$\log(n^n)$	+	−	+	−	+

+ yes
− no

Task 2

a) $T(n) = 2T(n/2) + n^4$

We can use the Master method

$f(n) = n^4$, $a=2$, $b=2$ $\quad n^{\log_2 2} = n \quad f(n) = \Omega(n^{\log_2 2 + 3})$

$2 \cdot f(\frac{n}{2}) \leq c \cdot f(n) \quad 2 \frac{n^4}{2^4} \leq c\, n^4 \quad c \geq \frac{1}{8}$

e.g. $c = \frac{1}{2}$ the inequality is true, so $\boxed{T(n) = \Theta(n^4)}$

b) $T(n) = T(7n/10) + n$

$f(n) = n$, $a=1$, $b = \frac{10}{7}$ $\quad n^{\log_{\frac{10}{7}} 1} = n^0 = 1 \quad f(n) = \Omega(n^{\log_{\frac{10}{7}} 1 + 1})$

$f(\frac{7n}{10}) \leq c \cdot f(n) \quad \frac{7n}{10} \leq c \cdot n \quad c \geq \frac{7}{10}$

e.g. $c = \frac{8}{10} = \frac{4}{5}$ $\quad \boxed{T(n) = \Theta(n)}$

c) $T(n) = 16\,T(n/4) + n^2$

$f(n) = n^2$, $a=16$, $b=4$ $\quad n^{\log_4 16} = n^2 \quad f(n) = \Theta(n^{\log_4 16}) = \Theta(n^2)$

~~$T(n) = \Theta(n^2 \lg n)$~~ $\quad \boxed{T(n) = \Theta(n^2 \lg n)}$

d) $T(n) = 7T(n/3) + n^2$

$f(n) = n^2$ $\quad a=7$ $\quad b=3$ $\quad n^{\log_3 7} \quad \log_3 7 \approx 1{,}7713$

As ε let's take $2 - \log_3 7$, so $f(n) = \Omega(n^{\log_3 7 + \varepsilon})$

$7 \cdot f(\frac{n}{3}) \leq c \cdot f(n) \quad 7 \cdot \frac{n^2}{9} \leq c \cdot n^2 \quad c \geq \frac{7}{9}$

e.g. $c = \frac{4}{5}$ $\quad \boxed{T(n) = \Theta(n^2)}$

e) $T(n) = 7T(n/2) + n^2$

$f(n) = n^2$ $\quad a=7$, $b=2$ $\quad \cancel{n^{\log}}$ $\log_2 7 \approx 2{,}8074$

As ε let's take $\log_2 7 - 2$, so $f(n) = O(n^{\log_2 7 - \varepsilon})$

$\boxed{T(n) = \Theta(n^{\log_2 7})}$

f) $T(n) = 2T(n/4) + \sqrt{n}$

$f(n) = n^{\frac{1}{2}}$ $\quad a=2$ $\quad b=4$ $\quad f(n) = \Theta(n^{\log_4 2}) = \Theta(n^{\frac{1}{2}})$

$\boxed{T(n) = \Theta(n^{\frac{1}{2}} \lg n)}$

g) $T(n) = T(n-2) + n^2 = T(n-4) + (n-2)^2 + n^2 = T(n-6) + (n-4)^2 +$
$+ (n-2)^2 + n^2 = \dots = T(0) + 2^2 + \dots + (n-6)^2 + (n-4)^2 + n^2 =$
$= T(0) + \sum_{i=0}^{n/2} (n-2i)^2 = T(0) + \sum_{i=0}^{n/2} n^2 - \sum_{i=0}^{n/2} 4ni +$
$+ \sum_{i=0}^{n/2} 4i^2 = \frac{n}{2} \cdot n^2 - \frac{4n^2}{2} + 4\frac{n}{2} = \frac{n^3}{2} - 2n^2 + 2n$

$\boxed{T(n) = \Theta(n^3)}$

Task 3

a) $T(n) = 4T(n/3) + n \lg n$

By Master theorem $a=4$, $b=3$ $f(n) = n \lg n$

$n^{\log_3 4}$, $\log_3 4 \approx 1{,}26186 \Rightarrow \log_3 4 > 1 \Rightarrow$

$\Rightarrow \boxed{T(n) = \Theta(n^{\log_3 4})}$

b) $T(n) = 3T(n/3) + n/\lg n$

By master theorem $a=3$, $b=3$, $f(n) = \frac{n}{\lg n}$

$n^{\log_3 3} = n^1$ $\log_3 3 = 1$ ~~..~~, ~~..~~

$\frac{1}{\lg n} = (\lg n)^{-1} \Rightarrow \boxed{T(n) = \Theta(n \lg \lg n)}$

c) $T(n) = 4T(n/2) + n^2\sqrt{n} = 4T(n/2) + n^{5/2}$

By master theorem $a=4$, $b=2$, $f(n) = n^{5/2}$

$n^{\log_2 4} = n^2$ $f(n) = \Omega(n^{\log_2 4 + 0{,}5})$

$2, f\left(\frac{n}{2}\right) \le c, f(n)$

$2, \frac{n^{5/2}}{2^{5/2}} \le c, n^{5/2}$ $c \ge \frac{2}{2^{5/2}}$

e.g. $c = \frac{2}{5}$ $T(n) = \Theta(n^{5/2}) = \Theta(n^{2{,}5})$

d) $T(n) = 3T(n/3 - 2) + \frac{n}{2}$

~~$\frac{n}{3} - 2$ is~~ The difference between $\frac{n}{3}$ and $\frac{n}{3} - 2$ is very small for sufficiently large n, so we can use Master method $a=3$, $b=3$

$f(n) = \frac{n}{2}$, $\log_3 3 = 1$ $f(n) = \Theta(n)$, so

$$\boxed{T(n) = \Theta(n \lg n)}$$

e) $T(n) = 2T(n/2) + n/\lg n$

By Master method $a=2$, $b=2$, $f(n) = \frac{n}{\lg n}$

$\log_2 2 = 1$, $\frac{1}{\lg n} = (\lg n)^{-1}$ $\Rightarrow$

$$\boxed{T(n) = \Theta(n \lg \lg n)}$$

f) $T(n) = T(n/2) + T(n/4) + T(n/8) + n$

let's use substitution method we want that

$T(k) \le ck$, when $k<n$, then

$$T(n) = T\left(\frac{n}{2}\right) + T\left(\frac{n}{4}\right) + T\left(\frac{n}{8}\right) + n \le$$

$$\le \frac{cn}{2} + \frac{cn}{4} + \frac{cn}{8} + n \le \frac{7}{8}cn + n$$

So, $\boxed{T(n) = O(n)}$

g) $T(n) = T(n-1) + \frac{1}{n}$

$$T(n) = T(n-1) + \frac{1}{n} = T(n-2) + \frac{1}{n} + \frac{1}{n-1} =$$

$$= T(n-3) + \frac{1}{n-2} + \frac{1}{n-1} + \frac{1}{n} = \dots = T(0) + \sum_{k=1}^{n} \frac{1}{k}$$

$$\sum_{k=1}^{n} \frac{1}{k} \le 1 + \int_1^n \frac{1}{x}\,dx = 1 + \log n - \log 1 = 1 + \log n$$

So $\boxed{T(n) = \Theta(\log n) \quad (\text{or } \Theta(\lg n))}$

h) $T(n) = T(n-1) + \lg n$

$T(n) = T(n-1) + \lg n = T(n-2) + \lg(n-1) + \lg n =$

$= T(n-3) + \lg(n-2) + \lg(n-1) + \lg n = \dots =$

$= T(0) + \lg 1 + \dots + \lg(n-2) + \lg(n-1) + \lg n =$

$= T(0) + \lg(n!)$

~~but we know, that lg(n!)~~ we know that

So $T(n) = \Theta(\lg n!)$, but $\Theta(\lg n!) = \Theta(n \lg n)$

(Stirling's approximation)

So $\boxed{T(n) = \Theta(n \lg n)}$

i) $T(n) = T(n-2) + 1/\lg n = T(n-4) + \frac{1}{\lg(n-2)} + \frac{1}{\lg n} =$

$= T(n-6) + \frac{1}{\lg(n-4)} + \frac{1}{\lg(n-2)} + \frac{1}{\lg n} = \dots =$

$= T(0) + \frac{1}{\lg 2} + \dots + \frac{1}{\lg(n-4)} + \frac{1}{\lg(n-2)} + \frac{1}{\lg n} =$

$= T(0) + \sum_{i=1}^{n/2} \frac{1}{\lg(n-2i)} \leq T(0) + \sum_{i=1}^{n/2} \frac{1}{a} =$

~~As we know~~ where $a = \lg 2$ if n is even or $a = \lg 3$ if n is odd

$= T(0) + \frac{n}{2} \cdot \frac{1}{a} \Rightarrow$ ~~$T(n) = \dots$~~

$\boxed{T(n) = \Theta(n)}$

j) $T(n) = \sqrt{n}\, T(\sqrt{n}) + n$

We can use Master theorem, if we will make some appointments: ~~$n = 2^k$, $\sqrt{n} = 2^{k/2}$, $k = \log n$~~

$n = 10^k$ $\sqrt{n} = 10^{k/2}$ $k = \lg n$

$T(10^k) = 10^{k/2}\, T(10^{k/2}) + 10^k$ | dividing by 10^k

$\frac{T(10^k)}{10^k} = \frac{T(10^{k/2})}{10^{k/2}} + 1$, let $y(k) = \frac{T(10^k)}{10^k}$

$y(k) = y\left(\frac{k}{2}\right) + 1$

Now let's use ~~Master's~~ theorem : $a=1$, $b=2$

$f(k) = 1 = k^0$, $\log_2 1 = 0 \Rightarrow$

$T(k) = \Theta(\lg k)$

$T(10^k) = 10^k \lg k$, $n = 10^k$ and $k = \lg n$,

$\boxed{\text{so } T(n) = \Theta(n \lg\lg n)}$

Task 4

a) Build-Max-Heap (A)

A.heap-size = A.length

for $i = \lfloor A.length/2 \rfloor$ down to 1

Max-Heapify (A, i)

Build_max-Heap' (A)

A.heap-size = 1

for i=2 to A.length

Max-Heap-Insert (A, A[i])

These procedures don't create the same heap when run on the same input array. Let's say we have A = {6, 4, 2, 8, 10}, With Build-Max-Heap we'll have {10, 8, 2, 4, 6}. With Build-Max-Heap' we'll have {10, 8, 2, 6, 4}.

b) For each insert we will have at most $O(\lg(n))$ time, and we are doing it n (n-2) times, so it'll be $\Theta(n \lg n)$.

Task 5

We know that $\lg k$ is monotonically increasing. Also we know, that when a summation has the form $\sum_{k=m}^{n} f(k)$, where $f(k)$ is monotonically increasing function, we can approximate it by integrals:

$$\int_{m-1}^{n} f(x)dx \le \sum_{k=m}^{n} f(k) \le \int_{m}^{n+1} f(x)dx$$

So we can say $\lg(n!) = \sum_{k=1}^{n} \lg k$, so

$$\int_{0}^{n} \lg x\, dx \le \sum_{k=1}^{n} \lg k \le \int_{1}^{n+1} \lg x\, dx$$

$$\int_{0}^{n} \lg x\, dx = \int_{0}^{n} \lg x \cdot 1\, dx = \lg x \cdot x - \int_{0}^{n} x \cdot \frac{1}{x \cdot \ln 2} dx =$$

$u = \lg x$ $\quad v' = 1$, $v = x$

$$= n \cdot \lg n - \frac{n}{\ln 2}$$

$$\int_{1}^{n+1} \lg x\, dx = (n+1)\lg(n+1) - \frac{n}{\ln 2}$$

So $\sum_{k=1}^{n} \lg k = \Theta(n \lg n)$