Problem 1

a) Let $f(n) = n^3 - 21n^2 + 100$. For $c = 101$ and $n_0 = 1$, we have for all $n \geqslant n_0$, $f(n) \leqslant cn^3$.

Moreover, $f(n) = n^2(n-21) + 100$ so for all $n \geqslant 21$, $f(n) \geqslant 0$.

So, for $c = 101$ and $n_0 = 21$, we have for all $n \geqslant n_0$, $0 \leqslant f(n) \leqslant cn^3$.

b) Let $f(n) = (n+10)^3$ and $g(n) = n^3$. Let us prove separately that $f(n) \in \Omega(g(n))$ and $f(n) \in O(g(n))$.

First, $(n+10)^3 \geqslant n^3$, so $f(n) \in \Omega(g(n))$. Second, for $n_0 = 10$ and for all $n \geqslant n_0$, $f(n) = (n+10)^3 \leqslant (2n)^3 = 8n^3$. This implies $f(n) \in O(g(n))$.

c) Let $c > 0$ be any constant. Let n_0 be an integer such that $1000/c < \log(n_0)$, *e.g.* $n_0 = \lfloor 2^{1000/c} \rfloor + 1$ where $\lfloor r \rfloor$ is the greatest integer lesser or equal to r. Then for all $n \geqslant n_0$, $0 \leqslant 1000n < cn\log(n)$.

d) On the one hand, $n! = n(n-1)\cdots 2 \cdot 1$ and on the other hand $n^n = \underbrace{n \cdots n}_{n \text{ times}}$.

Therefore $nn! \leqslant n^n$ for all positive integer n.

Let $c > 0$ be any constant and n_0 be any integer such that $1/n_0 < c$, *e.g.* $n_0 = \lfloor 1/c \rfloor + 1$. Then for all $n \geqslant n_0$, we get $0 \leqslant n! \leqslant 1/nn^n < cn^n$.

e) Notice that $P(r) = \sum_{i=2^{r-1}+1}^{2^r} 1/i \geqslant \sum_{i=2^r+1}^{2^{r+1}} 1/2^{r+1} \geqslant 1/2$. Therefore $H(2^r) \geqslant \sum_{i=1}^{r} P(r) \geqslant r/2$ and $H(n) \in \omega(1)$.

Problem 2

a) $-1 \leqslant \cos(2n) \leqslant 1$ so for all n, $3n^3 \leqslant f(n) \leqslant 7n^3$ and $f(n) \in \Theta(n^3)$. Using the maximum rule on Θ, we get $g(n) \in \Theta(n^3)$. Therefore $f(n) \in \Theta(g(n))$ (use for instance $f(n) = \Theta(n^3)$ and $n^3 = \Theta(g(n))$).

b) Consider the ratio $f(n)/g(n) = \log(n)^3/n$ and let us show that its limit is 0 when $n \to \infty$.

For this matter, we will prove the more general result : for any $\varepsilon > 0$, $\log(n)/n^\varepsilon \to_{n\to\infty} 0$.

Using L'Hopital's rule, we get $\lim_{n\to\infty} \frac{\log(n)}{n^\varepsilon} = \lim_{n\to\infty} \frac{1/n}{\varepsilon n^{\varepsilon-1}} = \lim_{n\to\infty} \frac{1}{\varepsilon n^\varepsilon} = 0$.

Now we can use this result : $\log(n)^3/n = (\log(n)/n^{1/3})^3 \xrightarrow[n\to\infty]{} 0$.

c) Consider the ratio $f(n)/g(n) = (n)^{\frac{1}{100}}/(\log n)^2$ and show that its limit is ∞ when $n \to \infty$ as before.

Problem 3

a) False. Consider $f(n) = \begin{cases} 1 & \text{if } n \text{ even} \\ n & \text{otherwise} \end{cases}$ and $g(n) = 1$.

b) True. Consider the ratio $r(n) := f(n)/g(n)$ which is defined for all n since g is a positive function. By definition of O, there exists $c > 0$ and n_0 such that for all $n \geqslant n_0$, $r(n) \leqslant c$. It remains to take the maximum of c and of all the values of $r(n)$ for $n < n_0$ to get a universal bound.

c) False. Take $f(n) = \log(n)$ and $g(n) = 2\log(n)$. Then $2^{f(n)} = n$ and $2^{g(n)} = n^2$.

d) True. $h(n) \in \Theta(g(n))$ implies $g(n) \in \Theta(h(n))$. Use $f(n) \in \Theta(g(n))$ to get $f(n) \in \Theta(h(n))$. Multiply by the well-defined function $1/h(n)$ to get $f(n)/h(n) \in \Theta(1)$.

e) Let $h(n) = \max(f(n), g(n))$ and $l(n) = \min(f(n), g(n))$. Then $\frac{f(n)g(n)}{f(n)+g(n)} = \frac{l(n)h(n)}{l(n)+h(n)}$. Since $h(n)+l(n) \in \Theta(h(n))$, we get $\frac{f(n)g(n)}{f(n)+g(n)} \in \Theta(l(n))$ as claimed.

Problem 4

a) Let $n = 2^r$. Then

$$f(n) = \sum_{i=0}^{r} 4^i \left(\frac{2^r}{2^i}\right)^{\theta} = 2^{r\theta} \sum_{i=0}^{r} (4/2^{\theta})^i = \begin{cases} 2^{r\theta}(r+1) & = & n^2(\log(n)+1) & \text{if } \theta = 2 \\ 2^{r\theta}\frac{1-(1/2)^{r+1}}{1-1/2} & = & n^3(2-1/n) & \text{if } \theta = 3 \end{cases}.$$

b) If $\theta = 2$ then $g(n) \in \Theta(n^2 \log(n))$.
If $\theta = 3$ then $g(n) \in \Theta(n^3)$.

c) Since $f(n)$ is increasing, we get $f(2^{r-1}) \leqslant f(n) \leqslant g(n) = f(2^r)$ where $r = \lceil \log_2(n) \rceil$. Therefore $f(2^{r-1})/f(2^r) \leqslant f(n)/g(n) \leqslant 1$. Since $f(2^{r-1})/f(2^r) \in \Theta(1)$ in both cases $\theta = 2, 3$, we conclude $f(n) = \Theta(g(n))$ and

$$f(n) \in \begin{cases} \Theta(n^2 \log(n)) & \text{if } \theta = 2 \\ \Theta(n^3) & \text{if } \theta = 3 \end{cases}.$$

Problem 5

1. Denote by s_i the value of s after i loop iteration. We take $s_0 = n$. Assume that our program never ends. Let i be an integer. If s_i is even then $s_{i+1} = s_i/2$ and $s_{i+2} \leqslant s_{i+1} + 1 = s_i/2 + 1$. If s_i is odd then $s_{i+2} = (s_i+1)/2 \leqslant s_i/2 + 1$.

   We know that $s_i > 2$ because otherwise, the program would stop at this step or the next step. Now, if $s_i > 2$, then $s_{i+2} \leqslant s_i/2 + 1 < s_i$. So the sequence $(s_{2n})_{n\in\mathbb{N}}$ is a strictly decreasing sequence of positive integer. Contradiction.

2. If $s_i \leq 3$ then $s_{i+2} \leqslant s_i/2 + 1 \leqslant s_i(1/2 + 1/3) = (5/6)s_i$. Consider the sequence $s_0, s_2, \ldots, s_{2k}, \ldots$ and let s_{2k} be the first element of this sequence such that $s_{2k} < 3$. Then

$$3 \leqslant s_{2(k-1)} \leqslant (5/6)s_{2(k-2)} \leqslant \cdots \leqslant (5/6)^{k-1}s_0 = (5/6)^{k-1}n$$

and so $(\log(3) - \log(n)) \leqslant (k-1)\log(5/6)$. The real $\log(5/6)$ is negative, so we get

$$k - 1 \leqslant \frac{(\log(3) - \log(n))}{\log(5/6)} = \frac{(\log(n) - \log(3))}{\log(6/5)} \in \Theta(\log(n)).$$

Now, either $s_{2k} \leqslant 1$ or $s_{2k} = 2$ and we need at most one more iteration for the program to terminate. Finally, the total number of iteration is lesser than or equal to $2k + 1 \in \Theta(\log(n))$.

Problem 6

Let $\mathcal{C}(A[1 \ldots n])$ be the cost of algorithm `max-element` on input A. Then,

$$\mathcal{C}(A[1 \ldots n]) = \begin{cases} d_1 & \text{if } n = 1 \\ d_2 + \mathcal{C}(A[2 \ldots n]) & \text{if } A[1] > \max - \text{element}(A[2 \ldots n]) \\ d_3 + 2\mathcal{C}(A[2 \ldots n]) & \text{otherwise} \end{cases}$$

for some integers $d_1 \leqslant d_2 \leqslant d_3$.

- Best case : Intuitively, $\mathcal{C}(A[1 \ldots n]) = d_2 + \mathcal{C}(A[2 \ldots n])$ is the best that can happen. By unrolling the recursion, one get

  $$\mathcal{C}(A[1 \ldots n]) = 2\Theta(1) + \mathcal{C}(A[3 \ldots n]) = \cdots = (n-1)\Theta(1) + \mathcal{C}(A[n \ldots n]) \in \Theta(n).$$

  It corresponds to input A sorted in decreasing order.

  To actually prove it, we proceed by induction. Let us prove that the best complexity on inputs of size n is attained for decreasing A :

  Initialization : All inputs of size 1 are "decreasing".

  Inductive step : The best complexity is obtained when $A[1] > \max - \text{element}(A[2 \ldots n])$: $\mathcal{C}(A[1 \ldots n]) = d_2 + \mathcal{C}(A[2 \ldots n])$. By recursion hypothesis, $A[2 \ldots n]$ must be decreasing to minimize the complexity. Therefore $A[1 \ldots n]$ is decreasing.

- Worst case : Similarly, we can prove that $\mathcal{C}(A[1 \ldots n]) = d_3 + \mathcal{C}(A[2 \ldots n])$ leads to the worst case complexity. Then,

  $$\mathcal{C}(A[1 \ldots n]) = (1+2)d_3 + 4\mathcal{C}(A[4 \ldots n]) = \cdots = (1+2+\cdots+2^{n-1})d_3 + 2^n\mathcal{C}(A[n \ldots n]) = \Theta(2^n).$$

  This complexity is attained for increasing A.