

Topic 1 · Sheet A: Worked answers and mark scheme

Asymptotic notation: O , Ω , Θ , o and ω

Suggested time

About 75 minutes

Total marks

73

Questions

12

Builds on

Nothing earlier

How marks are awarded. Each mark has a code, as on most UK exam papers.

B A mark for a correct statement or value on its own. It does not depend on any other mark.

M A method mark, for a correct approach, even if the final answer is wrong.

A An accuracy mark, for the correct answer *after* a correct method. It cannot be awarded without the M mark before it.

E An explanation mark, for a clear reason in words. A correct reason that is worded differently scores in full.

For true or false questions, the mark is for the correct decision; where a reason is asked for, it must be the right reason.

1 Quick decisions

4 marks

(a) **True.** (b) **False.** (c) **True.** (d) **True.**

B1 each part correct

4

Why this question These are the patterns you will use all the time: constants and lower-order terms do not change the growth rate, but a higher power always wins.

2 Finding the constants

4 marks

(a) $4n + 9 \leq 5n$ exactly when $n \geq 9$. So $c = 5$, $N = 8$ works (any $N \geq 8$ does).

(b) $n^2 - 3n \geq \frac{1}{2}n^2$ exactly when $\frac{1}{2}n^2 \geq 3n$, that is $n \geq 6$. So $c = \frac{1}{2}$, $N = 5$ works.

B1 (a) valid c and N

1

M1 (a) check, e.g. $4n + 9 \leq 5n \iff n \geq 9$

1

B1 (b) valid c and N

1

M1 (b) check

1

Common mistake Giving constants without checking them. Many pairs work, but you must show that your pair does, for every $n > N$.

3 A full Θ proof

6 marks

- (a) For $n \geq 1$ we have $5n \leq 5n^2$ and $3 \leq 3n^2$, so $f(n) \leq 2n^2 + 5n^2 + 3n^2 = 10n^2$. So $c_2 = 10$ works.
- (b) Every term is positive, so $f(n) \geq 2n^2$. So $c_1 = 2$ works.
- (c) $f(n) = \Theta(n^2)$, with $c_1 = 2$, $c_2 = 10$ and $N = 1$.

M1	(a) bounding each lower-order term by a multiple of n^2	1
M1	(a) adding the bounds	1
A1	(a) a valid c_2	1
M1	(b) dropping positive terms	1
A1	(b) a valid c_1	1
B1	(c)	1

Why this question Upper bound: make every term look like n^2 . Lower bound: throw away positive terms. These two moves prove almost every polynomial Θ bound.

4 Proofs by contradiction

6 marks

- (a) Suppose there are $c, N > 0$ with $n^2 \leq c \cdot 100n$ for all $n > N$. Dividing by $n > 0$ gives $n \leq 100c$ for all $n > N$. Take an integer $n > \max\{N, 100c\}$: then $n > 100c$, a contradiction. **So n^2 is not $O(100n)$.**
- (b) $2^{n+1} = 2 \cdot 2^n$, so $c = 2$, $N = 1$ works.
- (c) Suppose $4^n = 2^{2n} \leq c \cdot 2^n$ for all $n > N$. Dividing by 2^n gives $2^n \leq c$ for all $n > N$. An integer $n > \max\{N, \log_2 c\}$ gives $2^n > c$, a contradiction. **So 2^{2n} is not $O(2^n)$.**

M1	(a) assume c and N exist	1
M1	(a) divide by n to get $n \leq 100c$	1
A1	(a) choose $n > \max\{N, 100c\}$	1
B1	(b)	1
M1	(c) reduce to $2^n \leq c$	1
A1	(c) contradiction	1

Common mistake Thinking $2^{2n} = O(2^n)$ because “constants in the exponent do not matter”. A constant *added* to the exponent is a constant factor; a constant *multiplying* the exponent is not.

5 Using limits

6 marks

- (a) $\frac{n \log n}{n^{1.5}} = \frac{\log n}{\sqrt{n}} \rightarrow 0$ (for example by l'Hôpital's rule). So $f(n) = o(g(n))$.

- (b) $\frac{3n^2 + n}{n^2} = 3 + \frac{1}{n} \rightarrow 3$, a positive constant. So $f(n) = \Theta(g(n))$.
- (c) $\frac{2^n}{n^{10}} \rightarrow \infty$, since an exponential beats any power. So $f(n) = \omega(g(n))$.

M1	(a) the ratio	1
A1	(a)	1
M1	(b) the ratio	1
A1	(b)	1
M1	(c) the ratio	1
A1	(c)	1

Why this question Limit 0 gives little- o , limit ∞ gives little- ω , and a finite positive limit gives Θ . This is usually the fastest test.

6 Ordering by growth

6 marks

- (a) 1000, $\ln n$ and $\log_2 n$, $\sqrt{n}$, $n \log_2 n$, n^2 , 2^n , 3^n .
- (b) $\log_2 n$ and $\ln n$. By change of base, $\ln n = \frac{\log_2 n}{\log_2 e}$, a constant multiple of $\log_2 n$. So each is Θ of the other.

B4	(a) correct order. <i>Lose 1 mark for each function out of place</i>	4
B1	(b) the pair	1
E1	(b) change of base gives a constant factor	1

Common mistake Putting 2^n and 3^n together. Their ratio $(3/2)^n \rightarrow \infty$, so $2^n = o(3^n)$. Only the *base of a logarithm* can be changed for free.

7 Simplifying

6 marks

- (a) $\Theta(n^3)$. (b) $\Theta(n^3)$. (c) $5 \log_2 n$, so $\Theta(\log n)$. (d) $\Theta(2^n)$. (e) $\log_9 n = \frac{1}{2} \log_3 n$, so the sum is $\frac{3}{2} \log_3 n$: $\Theta(\log n)$. (f) $2n + 1$, so $\Theta(n)$.

B1	each part correct	6
----	-------------------	---

Common mistake In (f), answering $\Theta(n^2)$ without expanding. Simplify the expression first; the n^2 terms cancel.

8 True or false, with reasons

6 marks

- (a) **True.** If $f \leq c_1g$ for $n > N_1$ and $g \leq c_2h$ for $n > N_2$, then $f \leq c_1c_2h$ for all $n > \max\{N_1, N_2\}$.
- (b) **True.** If $f \leq cg$ for $n > N$, then $g \geq \frac{1}{c}f$ for $n > N$, so $g = \Omega(f)$ with constant $\frac{1}{c}$.
- (c) **False.** $\frac{2^n}{2^{n/2}} = 2^{n/2} \rightarrow \infty$, so 2^n is not $O(2^{n/2})$.

B1	(a) true	1
E1	(a) constants c_1c_2 and $\max\{N_1, N_2\}$	1
B1	(b) true	1
E1	(b) constant $1/c$	1
B1	(c) false	1
E1	(c) the ratio is unbounded	1

Why this question Part (a) is why you can chain bounds in an algorithm analysis, and (b) is why O and Ω are two views of the same comparison.

9 Little- o from the definition

6 marks

- (a) Given $c > 0$: $n \leq cn^2$ exactly when $n \geq \frac{1}{c}$. So $N = \frac{1}{c}$ works.
- (b) Take $c = 1$. Then $3n + 1 \leq n$ never holds, so **no N works for $c = 1$** and $3n + 1$ is not $o(n)$.
- (c) Suppose also $f \geq cg$ for $n > N_1$, with $c > 0$. Little- o with constant $\frac{c}{2}$ gives $f \leq \frac{c}{2}g$ for $n > N_2$. For $n > \max\{N_1, N_2\}$ we get $cg \leq \frac{c}{2}g$, so $g \leq 0$, contradicting $g > 0$. **So f is not $\Omega(g)$.**

M1	(a) solving $n \leq cn^2$	1
A1	(a) N in terms of c	1
M1	(b) choosing one c that fails	1
A1	(b)	1
M1	(c) using little- o with constant $c/2$	1
A1	(c) contradiction	1

Common mistake In (a), giving one fixed N . For little- o the condition is “for every c ”, so N must be allowed to depend on c .

10 Logarithm of a factorial

7 marks

- (a) $n! = 1 \cdot 2 \cdots n \leq n^n$, so $\log_2(n!) \leq n \log_2 n$. So $c = 1, N = 1$ works.
- (b) The largest $\lceil n/2 \rceil$ factors are each at least $\frac{n}{2}$, and the rest are at least 1, so $n! \geq (n/2)^{n/2}$. Taking logs, $\log_2(n!) \geq \frac{n}{2}(\log_2 n - 1)$. For $n \geq 4$, $\log_2 n - 1 \geq \frac{1}{2} \log_2 n$, so $\log_2(n!) \geq \frac{1}{4}n \log_2 n$. So $c = \frac{1}{4}, N = 3$.
- (c) $\log_2(n!) = \Theta(n \log n)$.
- (d) Both equal $2^{\log_2 3 \cdot \log_2 n}$, since $n = 2^{\log_2 n}$ and $3 = 2^{\log_2 3}$. **So they are equal.**

M1	(a) $n! \leq n^n$	1
A1	(a)	1
M1	(b) $n! \geq (n/2)^{n/2}$	1
M1	(b) taking logs	1
A1	(b) valid c and N	1
B1	(c)	1
B1	(d)	1

Why this question $\log(n!) = \Theta(n \log n)$ is the reason no comparison-based sort can beat $\Theta(n \log n)$. You will see it again with merge sort.

11 Exam style: a cubic with a negative term

8 marks

- (a) $f(n) \leq n^3 + 7 \leq 2n^3$ once $n^3 \geq 7$, that is $n \geq 2$. So $c_2 = 2, N = 1$.
- (b) $f(n) \geq \frac{1}{2}n^3$ exactly when $\frac{1}{2}n^3 - 10n^2 + 7 \geq 0$. For $n \geq 20$, $\frac{1}{2}n^3 \geq 10n^2$, so this holds. So $c_1 = \frac{1}{2}, N = 19$.
- (c) $f(n) = \Theta(n^3)$.
- (d) $\frac{f(n)}{n^3 \log n} = \frac{1 - \frac{10}{n} + \frac{7}{n^3}}{\log n} \rightarrow \frac{1}{\infty} = 0$. **Yes**, $f(n) = o(n^3 \log n)$.

M1	(a) dropping the negative term	1
A1	(a)	1
M1	(b) setting up $f(n) \geq c_1 n^3$	1
M1	(b) dealing with the $-10n^2$ term	1
A1	(b) valid c_1 and N	1
B1	(c)	1
M1	(d) the ratio	1
A1	(d)	1

Common mistake Using $c_1 = 1$ in (b). Then $f(n) \geq n^3$ needs $-10n^2 + 7 \geq 0$, which is false for every $n \geq 1$. Choose c_1 strictly below the leading coefficient.

12 Exam style: hidden polynomial

8 marks

- (a) $4^{\log_2 n} = (2^2)^{\log_2 n} = (2^{\log_2 n})^2 = n^2$.
- (b) $h(n) = n^2$, since $n \log_2 n$ grows more slowly than n^2 .
- (c) $f(n) \geq n^2$, and $n \log_2 n \leq n \cdot n$ since $\log_2 n \leq n$, so $f(n) \leq 2n^2$. So $c_1 = 1$, $c_2 = 2$, $N = 1$.
- (d) **Yes**, $f(n) = O(n^{2.5})$, since $f(n) \leq 2n^2 \leq 2n^{2.5}$. **No**, it is not $\Omega(n^{2.5})$, since $\frac{f(n)}{n^{2.5}} \leq \frac{2}{\sqrt{n}} \rightarrow 0$.

M1	(a) rewriting 4 as 2^2	1
A1	(a)	1
B1	(b)	1
M1	(c) lower bound	1
M1	(c) $\log_2 n \leq n$ for the upper bound	1
A1	(c) constants	1
B1	(d) O , with reason	1
B1	(d) not Ω , with reason	1

Why this question Exponents with logarithms often hide simple powers: $a^{\log_2 n} = n^{\log_2 a}$. Spotting this turns a strange-looking function into a polynomial.

Total: 73 marks. Spotted a mistake? Report it at unimixedtutor.com and we will fix it.