

Topic 5 · Sheet A: Worked answers and mark scheme

Divide and conquer: merge sort and recurrences

Suggested time

About 75 minutes

Total marks

73

Questions

12

Builds on

Topics 1 to 4

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 Running MERGE

4 marks

(a) $[1, 2, 3, 4, 7, 9]$. (b) **6**. (c) $1 \vee 2, 4 \vee 2, 4 \vee 3, 4 \vee 9, 7 \vee 9$, then 9 is copied: **5**. (d) $\Theta(k + t)$.

B1 each part correct

4

Why this question Each loop iteration places one entry of D in constant time. That linear cost is what makes the combine step cheap.

2 How many comparisons?

4 marks

(a) 1, 2, 3 are each compared with 4, then C is copied: **3**. (b) **5**. (c) $\min(k, t)$, when one array runs out after all its entries lose to the other array's first entry. (d) $k + t - 1$, since the last entry is always copied without a comparison.

B1 each part correct

4

Why this question Either way the number of comparisons is $\Theta(k + t)$ when k and t are about equal, which is all merge sort needs.

3 A full merge sort

6 marks

- (a) $[5, 2, 4, 7 \mid 1, 3, 2, 6]$, then $[5, 2 \mid 4, 7]$ and $[1, 3 \mid 2, 6]$, then single entries.
- (b) Bottom level: $[2, 5]$, $[4, 7]$, $[1, 3]$, $[2, 6]$. Next: $[2, 4, 5, 7]$, $[1, 2, 3, 6]$. Top: $[1, 2, 2, 3, 4, 5, 6, 7]$.
- (c) $4 + 2 + 1 = 7$.

B2	(a) every split correct. B1 for one level wrong	2
B1	(b) bottom level	1
B1	(b) middle level	1
B1	(b) top	1
B1	(c)	1

Common mistake Merging $[2, 4, 5, 7]$ and $[1, 2, 3, 6]$ and losing a duplicate 2. MERGE copies every entry; the output always has $k + t$ entries.

4 Setting up the recurrence

6 marks

- (a) $T(n) = T(\lfloor n/2 \rfloor) + T(n - \lfloor n/2 \rfloor) + \Theta(n)$ for $n > 1$, with $T(1) = \Theta(1)$.
- (b) $T(n) = 2T(n/2) + \Theta(n)$.
- (c) $T(2) = 2c + 2c = 4c$. $T(4) = 8c + 4c = 12c$. $T(8) = 24c + 8c = 32c$.
- (d) $T(n) = cn(\log_2 n + 1)$. It matches: $8c \cdot 4 = 32c$.

B1	(a) two recursive terms	1
B1	(a) $\Theta(n)$ combine and base case	1
B1	(b)	1
B1	(c)	1
B1	(c)	1
B1	(d)	1

Why this question Computing a few values and guessing is a perfectly good start. The next question shows how to prove the guess.

5 Unrolling

6 marks

- (a) $T(n) = 2T(n/2) + cn = 2(2T(n/4) + cn/2) + cn = 4T(n/4) + 2cn$. Each further step doubles the number of calls, halves their size, and adds another cn . After k steps: $T(n) = 2^k T(n/2^k) + kcn$.
- (b) Take $k = \log_2 n$, so $n/2^k = 1$: $T(n) = nT(1) + cn \log_2 n = cn(\log_2 n + 1)$.

(c) $\Theta(n \log n)$.

M1	(a) one substitution	1
M1	(a) a second substitution	1
A1	(a) general pattern	1
M1	(b) $k = \log_2 n$	1
A1	(b)	1
B1	(c)	1

Common mistake Writing $T(n) = 2T(n/2) + cn = 2 \cdot 2T(n/4) + cn$. Each substitution adds a new cn term as well; it is easy to drop one.

6 The recursion tree

6 marks

- (a) 2^i calls, each on $n/2^i$ entries.
- (b) $2^i \cdot \frac{n}{2^i} = n$.
- (c) Depths $0, 1, \dots, \log_2 n$: $\log_2 n + 1$ levels.
- (d) $n(\log_2 n + 1) = \Theta(n \log n)$.

B1	(a) number of calls	1
B1	(a) size	1
B1	(b)	1
B1	(c)	1
M1	(d) work per level times number of levels	1
A1	(d)	1

Why this question Merge sort does the same amount of work on every level of the tree, so the total is “work per level” times “number of levels”.

7 Good and bad splits

6 marks

- (a) Depth i has 3^i calls of size $n/3^i$, so n work per level, over $\log_3 n + 1$ levels. Total $n(\log_3 n + 1)$, which is $\Theta(n \log n)$.
- (b) $T(n) = T(n - 1) + n + 1$, so $T(n) = 1 + \sum_{m=2}^n (m + 1) = \frac{n(n+1)}{2} + n - 1$, which is $\Theta(n^2)$.

M1	(a) work per level	1
M1	(a) number of levels	1
A1	(a)	1
M1	(b) unrolling to a sum	1
A1	(b) the sum	1
A1	(b) $\Theta(n^2)$	1

Why this question Splitting into three instead of two only changes the base of the logarithm. Splitting off one entry destroys the benefit completely: you get back to a quadratic algorithm.

8 Why MERGE is correct

6 marks

- (a) At the start of iteration r , $D[1], \dots, D[r - 1]$ are in increasing order and are the $r - 1$ smallest entries of B and C together.
- (b) At $r = 4$: $D = [1, 2, 3]$, in increasing order, and 1, 2, 3 are the three smallest of 1, 4, 7, 2, 3, 9. It holds.
- (c) When the loop ends, $r = k + t + 1$, so all $k + t$ entries are in D in increasing order. That is the correct output.

B1	(a) increasing order	1
B1	(a) smallest $r - 1$ entries	1
M1	(b)	1
A1	(b)	1
M1	(c) invariant at the end	1
A1	(c)	1

Why this question The same pattern as Bubblesort: say what is true at the start of every iteration, then read off the result at the end.

9 Divide and conquer for the maximum

6 marks

- (a) $T(n) = T(\lfloor n/2 \rfloor) + T(\lceil n/2 \rceil) + \Theta(1)$, roughly $2T(n/2) + \Theta(1)$.
- (b) Each comparison is between two candidates and rules one of them out. We start with n candidates and end with one, so there are $n - 1$ comparisons.

(c) $\Theta(n)$.

(d) **No**: a scan also makes $n - 1$ comparisons. Divide and conquer does not help here.

M1	(a) two half-size calls	1
A1	(a) constant combine	1
E2	(b) each comparison eliminates one candidate	2
B1	(c)	1
B1	(d)	1

Why this question Divide and conquer pays off when combining is much cheaper than solving the whole problem directly. For the maximum it is not.

10 Exact comparisons in merge sort

7 marks

(a) $W(2) = 0 + 1 = 1$. $W(4) = 2 + 3 = 5$. $W(8) = 10 + 7 = 17$.

(b) Base case $n = 1$: $1 \cdot 0 - 1 + 1 = 0$. Inductive step: if $W(n/2) = \frac{n}{2} \log_2 \frac{n}{2} - \frac{n}{2} + 1$, then $W(n) = 2W(n/2) + n - 1 = n(\log_2 n - 1) - n + 2 + n - 1 = n \log_2 n - n + 1$.

(c) $W(1024) = 10\,240 - 1024 + 1 = 9217$. Bubblesort makes $\frac{1024 \cdot 1023}{2} = 523\,776$.

B1	(a)	1
B1	(a)	1
B1	(b) base case	1
M1	(b) substituting the hypothesis	1
A1	(b)	1
B1	(c) $W(1024)$	1
B1	(c) Bubblesort	1

Common mistake Using $\log_2 \frac{n}{2} = \frac{\log_2 n}{2}$. It is $\log_2 n - 1$.

11 Exam style: where the work is

8 marks

- (a) Depth i has 2^i calls of size $n/2^i$, each costing $(n/2^i)^2$, so the total is $\frac{n^2}{2^i}$.
- (b) Summing over all levels gives at most $n^2 \sum_{i \geq 0} 2^{-i} = 2n^2$. The top level alone costs n^2 . So $T(n) = \Theta(n^2)$.
- (c) Depth i costs 2^i , for $i = 0, \dots, \log_2 n$. Total $1 + 2 + \dots + n = 2n - 1$, which is $\Theta(n)$.

M1	(a) calls times cost per call	1
A1	(a)	1
M1	(b) geometric sum	1
A1	(b) upper bound	1
A1	(b) lower bound and conclusion	1
M1	(c) work per level	1
M1	(c) summing	1
A1	(c)	1

Why this question In (b) the top level dominates, so only the combine step matters. In (c) the leaves dominate, so only the number of base cases matters. Merge sort sits exactly in between.

12 Exam style: when n is not a power of 2

8 marks

- (a) $[5, 1 \mid 4, 2, 3]$. Left: $[5 \mid 1]$ merges to $[1, 5]$. Right: $[4 \mid 2, 3]$, with $[2 \mid 3]$ merging to $[2, 3]$, then $[2, 3, 4]$. Finally $[1, 2, 3, 4, 5]$.
- (b) Upper: $T(n) \leq T(2^\lambda) \leq c_2 2^\lambda \lambda \leq c_2(2n)(\log_2 n + 1) \leq 4c_2 n \log_2 n$ for $n \geq 2$, using $2^\lambda \leq 2n$ and $\lambda \leq \log_2 n + 1$. Lower: $T(n) \geq T(2^{\lambda-1}) \geq c_1 2^{\lambda-1}(\lambda - 1) \geq c_1 \frac{n}{2}(\log_2 n - 1) \geq \frac{c_1}{4} n \log_2 n$ for $n \geq 4$, using $2^{\lambda-1} \geq \frac{n}{2}$ and $\lambda - 1 \geq \log_2 n - 1$. So $T(n) = \Theta(n \log n)$.
- (c) Sizes $5 \rightarrow 3 \rightarrow 2 \rightarrow 1$: **3 levels**, and $\lceil \log_2 5 \rceil = 3$.

B1	(a) splits	1
B1	(a) merges	1
M1	(b) upper bound via $T(2^\lambda)$	1
A1	(b) upper bound in terms of n	1
M1	(b) lower bound via $T(2^{\lambda-1})$	1
A1	(b) lower bound in terms of n	1
B1	(c) levels	1
B1	(c) comparison	1

Why this question This is why we can ignore floors and ceilings: n is always within a factor 2 of a power of 2, and factors of 2 disappear inside Θ .

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