Assignment 5

Problem 1

(a)

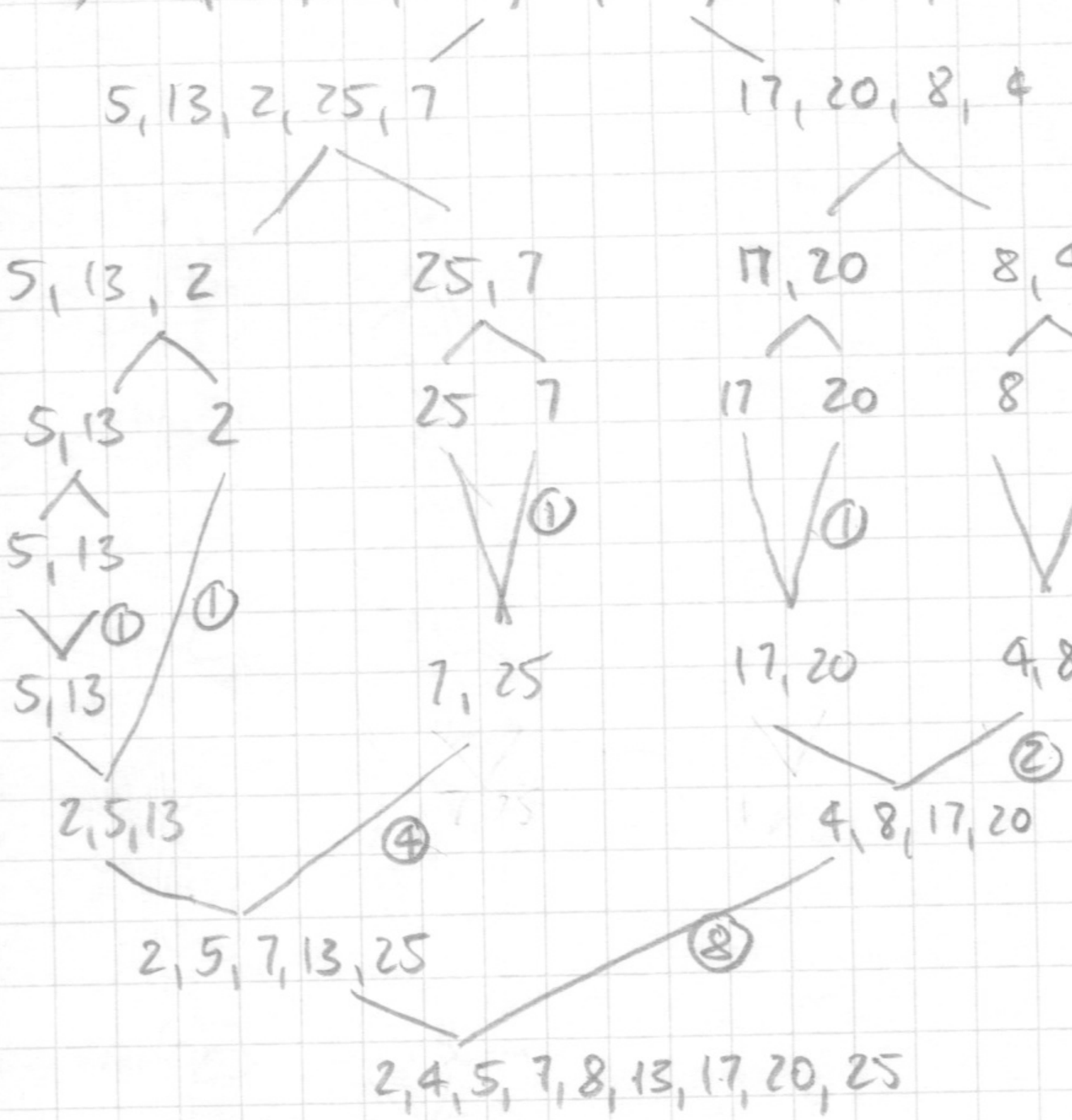
	gap	comp.
5, 13, 2, 25, 7, 17, 20, 8, 4	7	1
5, 13, 2, 25, 7, 17, 20, 8, 4		1
5, 4, 2, 25, 7, 17, 20, 8, 13	3	1
5, 4, 2, 25, 7, 17, 20, 8, 13		1
5, 4, 2, 25, 7, 17, 20, 8, 13		1
5, 4, 2, 25, 7, 17, 20, 8, 13		2
5, 4, 2, 20, 7, 17, 25, 8, 13		1
5, 4, 2, 20, 7, 17, 25, 8, 13		2
5, 4, 2, 20, 7, 13, 25, 8, 17	1	1
4, 5, 2, 20, 7, 13, 25, 8, 17		2
2, 4, 5, 20, 7, 13, 25, 8, 17		1
2, 4, 5, 20, 7, 13, 25, 8, 17		2
2, 4, 5, 7, 20, 13, 25, 8, 17		2
2, 4, 5, 7, 20, 13, 25, 8, 17		1
2, 4, 5, 7, 13, 20, 25, 8, 17		4
2, 4, 5, 7, 8, 13, 20, 25, 17		3
→ 2, 4, 5, 7, 8, 13, 17, 20, 25		

total 26 comparisons

2 marks for correct sorting
2 marks -"- counting comparisons

subtract 0.5 marks per mistake

(6) 5, 13, 2, 25, 7, 17, 20, 8, 4

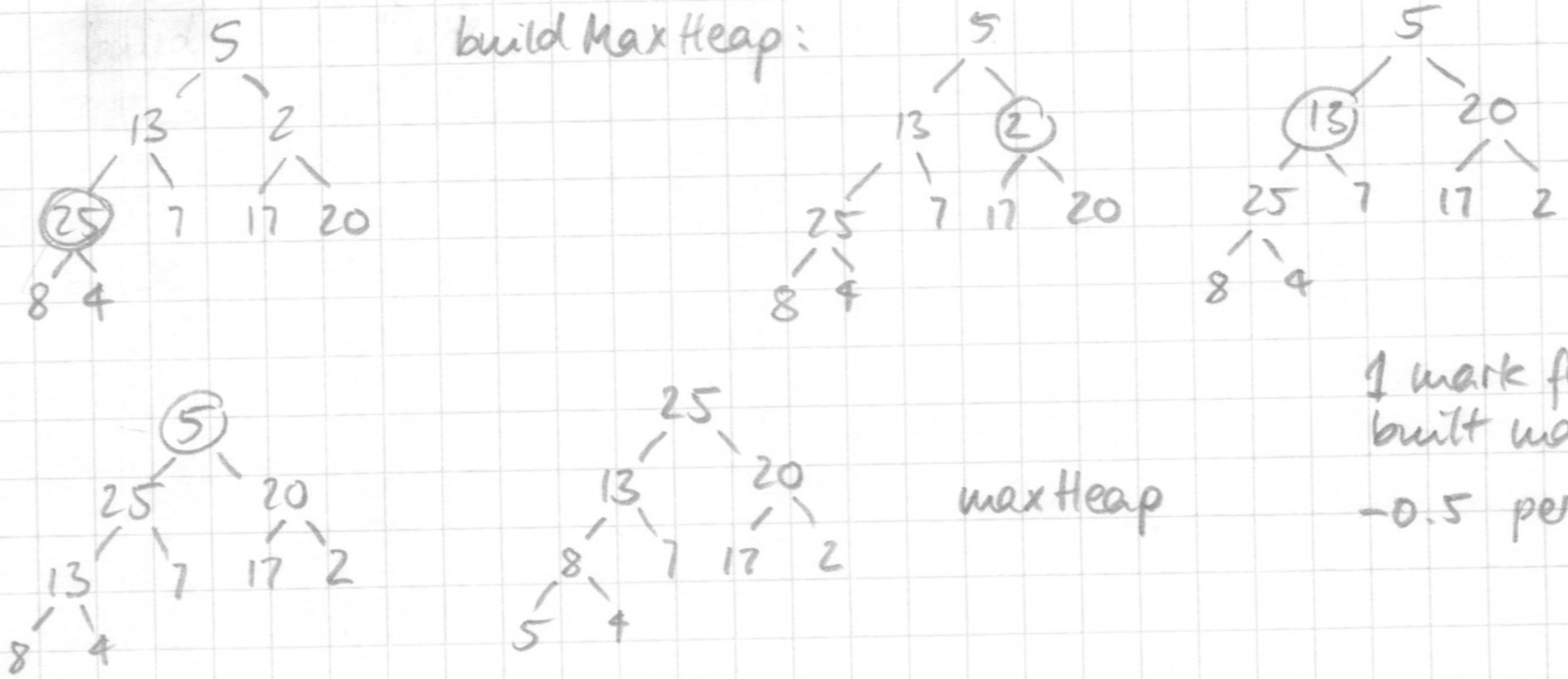


same marking rule as for (a)

total 19 comparisons

(c) 5, 13, 2, 25, 7, 17, 20, 8, 4

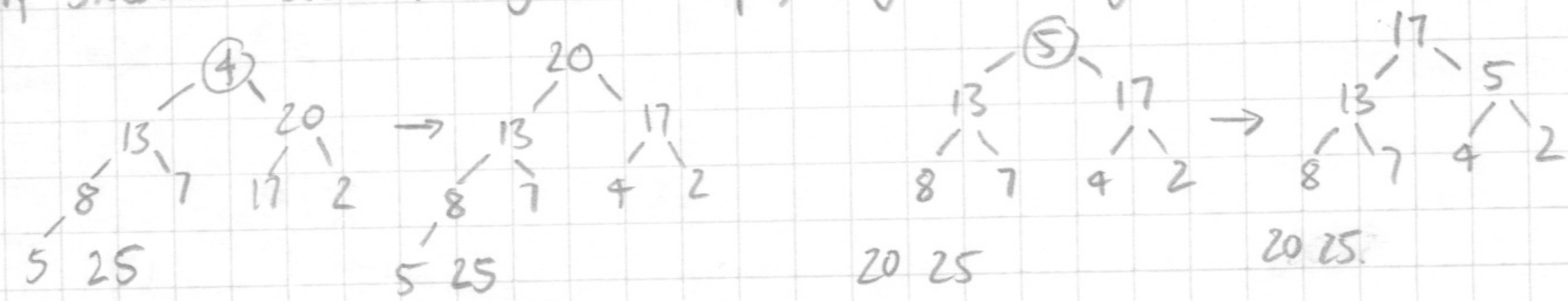
build MaxHeap:

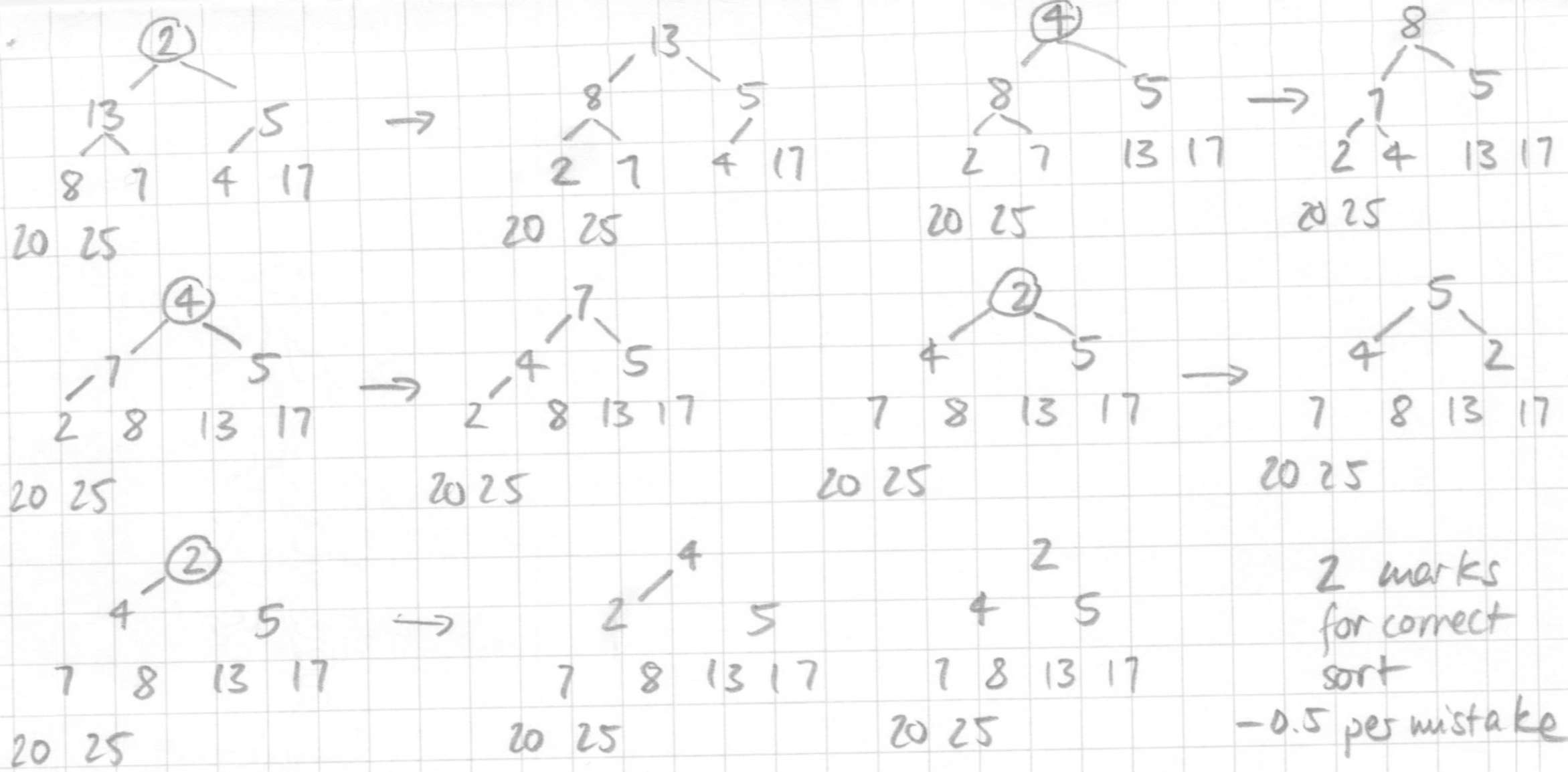


1 mark for correctly built maxheap

-0.5 per mistake

If students built wrong maxheap, they can still get marks for correct sorting:





2 marks for correct sort
-0.5 per mistake

Problem 2

(a) In every pass/round, the algorithm needs to make exactly one comparison. There are $N-1$ passes/rounds. The running time hence is $\Theta(N)$.
[1 mark for $\Theta(N)$, 1 mark for explanation]

(b) At best, we can assume the algorithm for merging always chooses the left of two equal elements first.
We then get $\Theta(\log N)$ many stages of merging.
In each stage, we need to make $\frac{N}{2} = \Theta(N)$ many comparisons.
$\Rightarrow$ The running time is $\Theta(N \log N)$.

[1 mark for $\Theta(N \log N)$, 2 marks for explanation]

Problem 3 [1 mark for each answer]

stable: Insertion Sort, Mergesort
not stable: Shellsort, Heapsort

Problem 4 [each 1 mark for correct answer (Θ-bound), 1 mark for explanation]

(a) $T(N) = 2T(\frac{N}{4}) + 1$

$a=2, b=4, f(N) = 1 = N^0, \quad z = \log_b a = \frac{1}{2} > 0$
Master Theorem (1) $\Rightarrow T(N) = \Theta(N^z) = \Theta(N^{\frac{1}{2}}) = \Theta(\sqrt{N})$

(b) $T(N) = 2T(\frac{N}{4}) + \sqrt{N}$

$a=2, b=4, f(N) = \sqrt{N} = N^{\frac{1}{2}}, z = \log_b a = \frac{1}{2}$
MT (2) $\Rightarrow T(N) = \Theta(N^z \log N) = \Theta(N^{\frac{1}{2}} \log N) = \Theta(\sqrt{N} \log N)$

(c) $T(N) = 2T(\frac{N}{4}) + N^2$

$a=2, b=4, f(N) = N^2, \quad z = \log_b a = \frac{1}{2} < 2$
Moreover, there is $n_0 \in \mathbb{N}$, $c<1$ such that $2 \cdot \frac{N^2}{4} \leq c \cdot N^2$ for all $N \geq n_0$, namely $c = \frac{1}{2}$, $n_0 = 1$.
MT (3) $\Rightarrow T(N) = \Theta(f(N)) = \Theta(N^2)$.

(d) $T(N) = 9T(\frac{N}{3}) + N$

$a=9, b=3, f(N) = N^1, \quad z = \log_b a = 2 > 1$
MT (1) $\Rightarrow T(N) = \Theta(N^z) = \Theta(N^2)$.

Problem 5 $T(N) = 2T(\frac{N}{2}) + N \log N$ [2 marks]

The problem is not that $f(N)$ is not equal to N^x for some x. $f(N)$ does not have to be equal to N^x for some x for the M.T. to apply!

$a=2, b=2, z=1. \quad f(N) = N \log N = \Omega(N^1) = \Omega(N^z)$

$f(N)$ is $\Omega(N^z)$, but there is no $x > z$ such that $f(N) = \Omega(N^z)$.
Therefore case (3) of M.T. does not apply.
Cases (1) and (2) don't apply, because $f(N) \neq O(N^z)$.