Problem 1 [10 marks]
Generally well done. Some students forgot to draw the intermediate heaps after each insertion or forget to draw the heaps for deleteMax.

Problem 2 [5+5=10 marks]

a) Some students built the heap using insertions and fix-up instead of using the bubble-down version of heapify (slide 22/25 Module 2)

b) Some students missed some details, not explaining how they do certain steps within their algorithms.

Problem 3 [5 marks]

• Some students only did bubble up or bubble down after swapping the element to delete with the originally last element.

• Some students did not include the pseudocode.

Problem 4 [5 marks]

• Some students only gave an array of specific length which cannot illustrate the comparisons needed is in $\Omega(n \log n)$.

• Many students’ proofs are actually for $O(n \log n)$, although they stated $\Omega(n \log n)$.

Problem 5 [3+2+4+1+1+3+3=17 marks]

a) Some students did not state an explicit value for $\lambda$.

c) • Some students forgot the number of comparisons needed for the insertion sort of the first $2\sqrt{m} + 1$ elements.

• Some students forgot the number of comparisons needed for partition.

e) Many students claimed that $\sqrt{m} + 2 \leq 2\sqrt{m}$ without stating that the reason is $m \geq 9$.

f) • Some students assumed $T(n)$ is an increasing function.

• Some students forgot to refer to part c).
• Some students forgot to refer to the first inequality in part e).

g) Using a) and f), deduce that $T(m) \in O(m^{3/2})$.

• Some students assumed $T(n)$ is an increasing function.
• Some students started from scratch, forgetting to apply the results from previous parts; part a), part f), and the second inequality in part e).

Problem 6  [5+9=14 marks]

a)  • Some students did not check if the left child and the right child exist before inserting them to the auxiliary heap
• Some students said insert and deleteMax takes $O(\log k)$ times without explaining the maximum size of B
• Some students did not include the pseudocode.
• Many students did not include a justification of correctness.

b) Some students’ codes did not compile. As stated in the assignment guideline, make sure your code compiles on the undergrad Linux environment before submitting it to MarkUs.