C/W One (Partial) Answers and Feedback

QUESTION 1

Answers

Part a: A small positive value.
E.g. 2 for
1 to load integer 0, 1 to store in memory

Part b: A small positive value.
E.g. 4 for
2 to load integer 2 and variable h, 1 to divide, 1 to store result

Part c: A small positive value.
e.g. **4 for**
2 to load integer 3 and variable k, 1 to divide, 1 to store result

Part d: Something with a fairly large constant, and a positive small multiple of n.
E.g. 100 + 2n
~100 to create the array object (allocate memory, etc.), 2 n to initialise each array element to 0

A frequent wrong answer was to just put 2 or 3 or similar.
This is too small and ignores n.
Note the answer is in the answers to Lab 2 (in bold face). People that did not do, or at least study, the labs, may have lost marks on this question.

Part e: A small positive constant.
E.g. 2 for
1 to get reference to A, 1 to store result

A fairly frequent wrong answer was to make it depend on n, because of "copying the elements" or similar. This is wrong because it is not copying the array at all, it is just assigning a pointer (or reference).
Note the answer is in the answers to Lab 2 (in bold face). People that did not do, or at least study, the labs, may have lost marks on this question.

QUESTION 2 (7 total)

Part a

Part I (3 marks)

If $2n^2 + n \; is \; O(n^2)$ then there exist constants c and n_0 such that $\forall n \geq \; n_0. \;\; 2n^2 + n \leq cn^2$

Rearrange inequality to find c:

$$2n^2 + n \leq cn^2$$
$$2 + \frac{1}{n} \leq c$$

As n tends to infinity 1/n tends to 0, so we will need c > 2.
For example, take $c = 3$ and the inequality is guaranteed to hold once n is big enough. Set $n_0 = 1$ and we're finished.

Errors seen:

- Incorrectly copying the expression for f(n) from the paper, into the definition of Big-Oh from the front page of the script. E.g. taking f(n) = n
- Algebraic errors in the manipulations.
- Identifying a value of n_0 but then substituting this into their equation, thus not proving $\forall n \geq n_0$.
- Confusion with dropping smaller terms; some drop smaller terms within the arithmetic which may leads them to a wrong answer, and is not "from the definition". The "drop smaller terms method" is a heuristic for working this out in your head, not for a proof "from the definition"

Part ii (3 marks)

If $f(n) \in \Omega(n^2)$ then there exist constants c and n_0 such that $\forall n \geq n_0,\ 2n^2 + n \geq cn^2$

Rearrange as before to find c which gives $2 + \frac{1}{n} \geq c$. If $c = 2$ then even if 1/n is almost 0 c will still be smaller. As such, set $n_0 = 1$ and we're done.

The cases
c = 1 n0 = 1
c = 2 n0 =1
work and prove the result.

Picking c= 3 is wrong, as it would require that $\forall n \geq n_0,\ n \geq n^2$ which cannot work for any n0.

Common errors are similar to those for part (i).

Part b (1 mark)

$f(n)$ is both $O(n^2)$ and $\Omega(n^2)$ so we can conclude that is it also $\Theta(n^2\ )$
No need for a full proof giving values for c', c''and n0 (as it is just one mark, and just asks for wha you can say.

QUESTION 3 (8 marks total, 2 per question)

Answers

Part a) – D

Part b) – C

Part c) – C

Part d) – A
The frequent wrong answer B was considered a “near miss” and given one mark.

QUESTION 4 – 4 marks total

Answers

Part a (1 mark)

The sort is **selection sort.**

Part b (2 marks)

Answer should show 3 and 2’ being compared and swapped, and then 2’ and 2 not being swapped.

Final state of the array should be [2’,2,3].

Part c (1 mark)

The sort is unstable – as the 2 and 2’ swap their order.

If an error in part b lead to sorting giving [2,2’3] then STABLE was also accepted (this is generous!)