2021 Algorithmics (HESS) external assessment report

General comments

Excellent responses were often seen in questions where students had to demonstrate an understanding of graphs or describe how a context could be modelled using abstract data types. In questions where students needed to describe algorithms, they often defined the inputs of the algorithm very clearly.

Students who had difficulty with applied questions relating to abstract concepts from Unit 4, such as Big-O notation, complexity classes, intractability and undecidability, often provided false definitions. It is critical that students understand the precise definitions of these concepts to draw appropriate conclusions.

Students are advised to:

- use more plain language and less programming code syntax to express their method in questions requiring the description of algorithms using pseudocode, such as Questions 3b., 4d. and 9c. in Section B
- ensure that their algorithm returns an appropriate value in all circumstances
- clearly indicate which part of their response relates to each aspect of the question in questions that ask for more than one item in the response, such as Questions 7 and 8ciii. in Section B
- keep their response focused on the particular concept/question described in the prompt for questions that ask to discuss or explain a concept, such as Questions 8, 9, 10 and 11 in Section B, and avoid adding extra facts that don't directly relate to the question prompt
- ensure that they understand the execution of the graph algorithms specified in Unit 3 Outcome 2 and can explain why they produce correct results. Many students had difficulty with this in Questions 1cii. and 5b. of Section B. The techniques of induction and contradiction that are studied in Unit 3 Outcome 3 can be applied to understand the correctness of many of these algorithms
- study Big-O notation in an abstract sense, not only its application to classifying the relationship between the size of an algorithm's input and its running. Misconceptions relating to this were seen in Questions 9 and 14 of Section A and Question 7 in Section B
- ensure their handwriting is clearly legible. Many responses require precise statements and the illegibility of even one word can significantly alter the meaning of a response.

Specific information

This report provides sample answers or an indication of what answers may have included. Unless otherwise stated, these are not intended to be exemplary or complete responses.

The statistics in this report may be subject to rounding resulting in a total more or less than 100 per cent.

Section A

Question	Correct Answer	%A	%B	%C	%D	Comment
1	A	74	9	11	6	
2	D	3	7	10	80	
3	B	49	47	3	1	Breaking the problem down into smaller parts by creating a truth table for the expressions may have assisted students in this question: • NOT (studying a language outside of school AND failing other subjects), and • NOT (studying a language outside of school OR failing other subjects)
4	B	39	57	1	3	To differentiate between breadth-first and depth-first search, students should consider whether the algorithm begins exploring new nodes before it finishes exploring the current node.
5	A	72	16	10	3	
6	D	1	14	30	55	Decomposing the PageRank formula into its parts may have assisted students in this question. • $\sum_{u \in U} \frac{PR(u)}{\text{out-degree}(u)}$ is the probability that v is chosen as the new node when the explorer chooses to follow an edge from their current node. • $\frac{1}{N}$ is the probability that v is chosen as the new node when the explorer chooses to randomly choose a new node. The d and $(1 - d)$ that these two expressions are multiplied by are the probabilities that each particular situation is chosen.
7	A	63	9	20	8	
8	B	2	68	13	17	
9	C	37	7	28	28	In the worst-case scenario, the pivot chosen by the QuickSort algorithm is the highest or lowest element. In this scenario, one sub-list is empty and the other is one element smaller than the original list.
10	A	68	5	17	10	
11	C	5	4	84	7	
12	A	64	9	2	5	
13	C	7	5	84	3	
14	C	4	39	21	36	From the prompt it is known that the running time is bounded at the top by $a \cdot n^4$ and at the bottom by $b \cdot n^3$. Any looser upper or lower boundaries are necessarily true. $O(n^2)$ and $O(n^3)$ are both tighter upper bounds. $\Omega(n^4)$ is a tighter lower bound. $\Omega(n^2)$ is a looser lower bound, and hence necessarily true.
15	D	5	18	10	66	
16	B	7	79	9	5	

Question	Correct Answer	%A	%B	%C	%D	Comment
17	C	4	7	76	13	
18	A	40	6	9	45	The time complexity depends only on the number of classes, c, because the time taken to determine whether or not a class has 25 students does not depend on the number of students in the class (you stop counting at 25).
19	B	1	86	6	7	
20	D	12	9	7	72	

Section B

Question 1a.

Marks	0	1	Average
%	3	97	1.0

The answer was 6.

Question 1b.

Marks	0	1	Average
%	14	86	0.9

The answer was 3. The degree of a node is a count, not a measurement of angle or temperature. Some students incorrectly measured their answer in the units of degrees.

Question 1ci.

Marks	0	1	Average
%	11	89	0.9

There are negative edge weights in the graph. Some students incorrectly stated there are negative cycles in the graph.

Question 1cii.

Marks	0	1	2	Average
%	63	26	11	0.5

Node B will have an incorrect distance. Dijkstra’s algorithm calculates the distance of B as 4. The correct shortest distance is 2, via the path A, C, D, B.

Many students identified other nodes and some students who correctly identified Node B identified the path A, D, B as shortest, with a cost of 3. Students are encouraged to step through the execution of Dijkstra’s algorithm in this kind of question.

Question 2a.

Marks	0	1	2	3	Average
%	10	19	38	33	2.0

An array of dictionaries would be suitable. Each dictionary in the array represents an addition of a chemical to a reaction, with the dictionary containing (key, value) pairs for each of the name, time and amount of chemical. This would allow the chemical reaction to be recreated because an array allows for accessing any part of the reaction and the dictionary stores all of the relevant data for each addition.

Many students provided suitable data models but did not provide a justification.

Question 2b.

Marks	0	1	2	Average
%	39	37	24	0.9

add_operation : ChemEvents × String × Integer × Integer → ChemEvents

get_operation : Integer × ChemEvents → String

Students should use the name of the abstract data type (ADT) given in the prompt within their signatures.

Question 3a.

Marks	0	1	2	Average
%	9	39	52	1.5

An undirected, weighted graph ADT in which each node represents a region on the board. Regions that are adjacent with a normal border are connected by an edge with weight 1 and regions that are adjacent with a river border are connected by an edge with weight 2. No edge would exist between adjacent regions with a mountain border.

Some of the responses were vague. Students are encouraged to explicitly capture the features of the context in their modelling.

Question 3b.

Marks	0	1	2	3	4	5	Average
%	27	19	16	18	14	5	1.9

A possible answer was:

```
let s be the source node and x be the destination node.
set s as visited and all other nodes as unvisited.
for i in V, initialise distance[i] to infinity and path[i] to null
initialise distance[s] = 0 and path[s] as a list containing s

while Q is not empty
  let u be the unvisited node with the lowest distance
  set u as visited
  for each neighbour v of u
```

```
        if distance[u] + weight(u,v) < distance[v]
          distance[v] = distance[u] + weight(u,v)
          path[v] = path[u] with v appended

   return path[x]
```

Students often provided some form of graph traversal algorithm. High-scoring responses were based on Dijkstra's algorithm. Low-scoring responses were often based on breadth-first or depth-first search and did not always account for the cost of traversing edges or determining the path to the destination node.

Question 3c.

Marks	0	1	2	3	Average
%	40	11	22	26	1.4

Samarth can use Dijkstra's shortest path algorithm to determine the cost from a source region to each other region and repeat this for all regions on the graph. He can then average the source-destination costs for each choice of source to determine the region with the lowest average cost.

The graphs for HexaQuesta game boards are not dense, with nodes having a maximum degree of six. Therefore, $|E| \propto |V|$. The running time of running Dijkstra's algorithm $|V|$ times would be $O(|V|^2 \log(|V|))$.

High-scoring responses typically applied the Floyd-Warshall algorithm and then averaged the cost to each destination for each choice of starting node. This approach had a running time $O(|V|^3)$. Low-scoring responses were vague in their description of how each choice of starting node was scored. Generally, students did not engage well with the efficiency aspect of this question.

Question 4a.

Marks	0	1	2	Average
%	7	28	66	1.6

The weights in the model could represent the time required to travel between two intersections in the warehouse. The direction would be important because the time to travel between two intersections could be asymmetric. That is, it might be hard to move from A to B if there are already robots moving from B to A; however, moving from A to B would be quick.

High-scoring responses clearly addressed both the weighted and directed aspects of the question, while low-scoring responses ascribed directions to the nodes of the graph.

Question 4b.

Marks	0	1	2	Average
%	26	42	32	1.1

Breadth-first search (BFS) will necessarily return a valid path, but it would not necessarily be the shortest path. This is because BFS will return a path with the fewest number of edges; however, because the model uses weighted edges, the shortest path could be one with a greater number of edges.

High-scoring responses clearly described why the path returned by BFS would not be valid due to the weighted edges in the graph.

Question 4c.

Marks	0	1	2	3	Average
%	13	22	14	51	2.1

The stack ADT only allows access to the top element in the stack. However, to determine whether a stack is unsafe requires access to all boxes in the pile. An array ADT would be appropriate as it allows access by index to each element.

Responses that scored highly provided appropriate reasons, based on how the swap operation would need to interact with the data, to explain why a stack was unsuitable and to justify the choice of a new ADT.

Question 4d.

Marks	0	1	2	3	Average
%	49	22	16	12	0.9

The answer was:

```
Let boxes be an array, with boxes[0] the bottom of the stack.

check_safe(boxes):
   if length of boxes < 2
         return True
   if boxes[0] ≥ boxes[1]
         return check_safe(tail(boxes))
   else:
         return False
```

Students often incorporated the comparison of adjacent items in the stack in their algorithm. Many students with recursive responses did not return the result of the recursive call. For example, they called it the check_safe method with appropriate arguments but did nothing with its return value. When attempting to compare adjacent items in the stack, some students instead compared the indices of the items. If the prompt requests a recursive algorithm, students should not provide iterative methods.

Question 4e.

Marks	0	1	2	3	4	Average
%	41	9	7	18	26	1.8

A possible answer was:

An unweighted, state graph with each node representing a possible pile arrangement and edges representing possible transitions between arrangements. An array could be used for each node to store the arrangement of the boxes. Once all states and the possible transitions between them have been generated, a single-source shortest path algorithm can be used to find the shortest path between the starting configuration of the pile and a safe configuration of the pile. The optimal flips required can be read off from the shortest path as the edges in the path represent the flips (transitions).

High-scoring responses clearly addressed both how a combination of ADTs could represent a state graph and then how the solution from a single-source shortest path (SSSP) algorithm could be used to solve the problem. Low-scoring responses described combinations of ADTs that were not fit for purpose, including responses that did not include a graph as any part of the solution despite the prompt indicating that a SSSP algorithm should be part of the solution.

Question 5a.

Marks	0	1	Average
%	70	30	0.3

The answer was edges. A commonly incorrect response was iterations.

Question 5b.

Marks	0	1	2	3	Average
%	67	15	13	4	0.6

The shortest path from the source to v with at most i edges will either have i edges or be the shortest with at most $(i-1)$ edges.

In the case where the shortest path has less than i edges, $distance[i,v] = distance[i-1,v]$, and this is the initial value set by the algorithm.

In the case where the shortest path has i edges, then it will end in an edge (u,v), and $distance[i-1,u]$ will be the shortest path from the source to u with at most $(i-1)$ edges. The algorithm tests every edge in the graph to see if there is a $distance[i-1,u] + weight(u,v)$ that is shorter than $distance[i-1,v]$ or other i length paths, and if found it updates the currently stored shortest distance.

The better responses attempted to systematically address the different cases for how the shortest path with at most i edges would be constructed and provided evidence that the algorithm would be correct for each case. Students should ensure that they develop their understanding of the underlying logic of the specified graph algorithms.

Question 6a.

Marks	0	1	2	Average
%	32	18	50	1.2

Step 2 in Eddie's process will not work because it requires checking whether an arbitrary program will terminate on a given input. This is a version of the Halting Problem, which is known to be an undecidable problem and so there can be no algorithm for it that will always give the correct answer.

Responses that recognised the correspondence with the Halting Problem scored highly.

Question 6b.

Marks	0	1	2	Average
%	42	14	43	1.0

Eddie could impose a time limit so that the user's program has to terminate within this time limit, otherwise it is deleted.

High-scoring responses provided improved processes that directly addressed the impossibility of verifying whether the code would terminate in the general case. Lower-scoring responses included processes that required users to provide code with a low asymptotic running time. Restricting the user's code to a lower-order model of computation was also a viable strategy, but this was not seen in student responses.

Question 7

Marks	0	1	2	3	4	Average
%	26	30	15	11	18	1.7

The best-case time complexity for this algorithm is $O(n)$. This occurs when the input is reverse sorted, such as $[5,4,3,2,1]$. The worst-case time complexity for this algorithm is $O(n^2)$. This occurs when the input is sorted, such as $[1,2,3,4,5]$.

Most students realised that sorted inputs would lead to the extreme cases for running time for inputs of a certain size. A common misconception was that best case time complexity corresponded to best running time, typically described as occurring for the input of an empty list. Students should ensure that they are considering how the running time of an algorithm changes relative to the nature of an input of size n.

Question 8a.

Marks	0	1	2	Average
%	41	32	27	0.9

The travelling salesman problem is in NP because given a candidate path, we can verify in linear time if the total weight of the path is below some value x.

High-scoring responses clearly described how it could be determined that the travelling salesman's problem was in NP by verifying that the path length met the requirement. Many students provided statements about the time taken to find solutions to the problem. These were both not relevant to the question and demonstrated a misconception that problems in NP were necessarily difficult to solve.

Question 8b.

Marks	0	1	2	Average
%	18	62	20	1.1

Starting at the source node, repeatedly add the lowest-cost edge connecting the most recently visited node and the set of unvisited nodes. When all nodes have been visited, add the edge from the most recently visited node back to the source node.

Most students provided appropriate greedy strategies, but many did not describe how a complete solution that returned to the starting node would be created.

Question 8ci.

Marks	0	1	2	3	Average
%	32	37	23	8	1.1

Each candidate solution is scored based on the total weight of the candidate path. If the score of the new candidate is lower than the current, then it is accepted. Otherwise, there is a chance that worse candidates will be randomly accepted. Worse candidates are less likely to be accepted as the temperature decreases and if the new candidate is a lot worse than the current.

Many students commented on the behaviour of the temperature and the effect the temperature would have on the likelihood of accepting worse solutions. High-scoring responses also addressed how the difference in scores between the candidates would influence the likelihood of accepting worse solutions.

Question 8cii.

Marks	0	1	Average
%	41	59	0.6

The algorithm could terminate after a fixed number of iterations.

Most students were able to provide an appropriate criterion based on the number of iterations, the temperature or the score of the current candidate. Two common incorrect responses were to terminate when a new candidate was accepted or when a valid solution was found.

Question 8ciii.

Marks	0	1	2	Average
%	23	26	52	1.3

A limitation is that simulated annealing is not guaranteed to find the optimal solution. The algorithm may either get stuck in a local minimum or it may terminate before it can find the optimal solution. An advantage is that simulated annealing is able to find reasonable solutions very quickly.

When a response to a question has two parts, such as an advantage and a limitation, students should ensure that both parts of their response are clearly identifiable.

Question 9a.

Marks	0	1	2	3	Average
%	11	20	48	21	1.8

Generate all possible sub-arrays of the profits array by considering arrays of length 1, then 2, and so on. For each of the sub-arrays, sum all of its houses to determine its profit. Select the sub-array that has the highest profit.

High-scoring responses provided detail about how the pool of candidate solutions would be generated and scored.

Question 9b.

Marks	0	1	2	3	Average
%	31	29	39	1	1.1

A divide-and-conquer approach could be used to find the greatest profit. The array of profits could be divided in half and then the highest profit of each half calculated. When merging the solutions, three possibilities would need to be considered: the highest profit in the left half, the highest profit in the right half, and the highest profit that spans both halves. To find the highest profit that spans both halves, each half would need to determine the highest profit it can return that joins to its neighbours.

Low-scoring responses did not adequately relate the algorithm design pattern to the particulars of the given problem.

Question 9c.

Marks	0	1	2	3	4	Average
%	60	18	14	4	3	0.7

The answer was:

```
Let P be the array of profits
Initialise best[0] = [P[0]]
for i = 1 to (length(P) – 1):
  best[i] = max(P[i], P[i] + best[i-1])
return max(best)
```

Most students provided algorithms that featured some elements of the dynamic programming design pattern. Many students were not able to identify an appropriate sub-structure to relate the solutions of sub-problems to the solutions of previous sub-problems.

Question 10a.

Marks	0	1	2	Average
%	14	44	41	1.3

A neural network does not need a pre-defined algorithm with all the factors that define ‘Zion’, but rather can be trained based on existing photographs and the knowledge of whether these photos are or aren’t of Zion.

High-scoring responses described motivations that were based in the specific context. Low-scoring responses described generic motivations for using neural networks.

Question 10b.

Marks	0	1	2	Average
%	21	40	39	1.2

Examples were:

- The training set of 100 photos is too small for a neural network.
- It is not appropriate to test a neural network on the same data set that was used for its training. This will just confirm whether it has learnt these photos, not whether it has learnt to recognise Zion.

When a prompt specifies a number of examples to be provided, students should ensure that they provide that number of examples and that each example is its own discrete point. Some students provided elaborate explanations for one problem with Zion’s current approach.

Question 11a.

Marks	0	1	2	3	Average
%	42	23	18	17	1.1

Manu's explanation is incorrect because an undecidable problem is one for which there is no single algorithm that can be used to solve every instance of such problems. However, it is possible that some instances of such problems can be solved.

Low-scoring responses often conflated the concepts of undecidability and the Halting Problem, including false statements similar to 'undecidable problems are problems for which no algorithm will ever halt'.

Question 11b.

Marks	0	1	2	3	Average
%	34	24	30	12	1.2

Undecidable problems are those for which no algorithm is possible. Intractable problems are those for which an algorithm is possible, but all algorithms are inefficient. Hence intractable problems are decidable. Even though it may be difficult to find a solution, it may still be possible to quickly check the solutions of an intractable problem.

Low-scoring responses often had relevant definitions for intractability but could not explain the errors in Raya's definition.

Question 11c.

Marks	0	1	2	Average
%	31	47	22	0.9

Strong AI describes the concept of artificial intelligence mechanisms that are able to go beyond mimicry of human behavior or language and that have human-like understanding and thinking processes.

High-scoring responses clearly indicated that the AI's understanding went beyond the appearance of understanding. A common incorrect response was that strong AI described artificial intelligences that were able to pass the Turing Test or that appeared human-like in intelligence.

Question 11d.

Marks	0	1	2	Average
%	62	9	29	0.7

Instead of being in a room, suppose that the person in the room has memorised the entire library of possible utterances and responses that were previously in the books. Now, upon receiving an utterance in Chinese, the person searches their memory for the appropriate response and replies with that response. The person appears to understand Chinese, but actually does not. Hence, the System's Reply does not provide evidence that the room as a whole should be treated as a system.

Some students were able to appropriately describe the narrative for the counterargument, but did not state the logic underpinning its conclusion.