COMP26120 Algorithms and Imperative Programming

David Rydeheard
Joshua Knowles
Milan Mihajlovic

Comments

Q1 This question was concerned with Dijkstra's algorithm and its applications. Most of the students wrote the algorithm correctly, but quite a few lost marks by forgetting to write its generalisation to the problem of finding the shortest path between all pairs of nodes in a weighted graph. The algorithm complexity was not a strong point, but some students did it right. The application of the algorithm to a concrete graph suffered mainly from the lack of detail in the progression of the priority queue used in the algorithm. The advanced part of the question was answered well, with the main oversights being handling of multiple hops in a single day. Overall, 171 students took this question, with 144 (84%) getting 10+ marks and 83 (48%) getting 14+ marks.

Q2 (Knapsack)

The first part, requiring a detailed explanation for the branch and bound approach to solving 0/1 knapsack, was answered very well by a few, and the majority got 50% or more of the 14 marks available. However, it was clear that some students had not revised this and gave a very poor attempt at answering - disappointing given that this was an optional question.

The second part asked for the most appropriate algorithm for solving each of two different instances of 0/1 knapsack, and an explanation for the choices. A few students attained full marks, but a lot lost several marks because the explanations merely repeated some details in the question rather than linking that to specific workings and behaviour of the algorithms.

Q3 This question was concerned with hashing. Most of the students did the first part right, where collisions are handled by linear probing. By contrast, double hashing was a much greater challenge, with most of the mistakes associated with wrong combinations of the primary and secondary hash function to obtain the final hash function. The main problem with the final part of the question (writing a pseudo code for finding a key in a hash table) was the lack of generality - the students frequently oversaw the significance of deactivated cells and circular search to obtain a correct algorithm. Overall, 194 students took this question, with 169 (87%) getting 10+ marks and 113 (58%) getting 14+ marks.

Q4. This question covered graph algorithms and basic facts about traversal techniques.

Whilst some students got good marks, the majority of answers were weak to poor, giving an overall average for the question of 50%.

It is clear that many students neither attended the lectures on this topic nor did proper revision from the notes and books. Most was standard material on DFS - code and properties. All second year CS student here should know this material. Some was on algorithm derivation using DFS for (a) counting the number of components. Some students did not understand the notion of a component of a graph(!); (b) 2-colourability: most student tried this part and many came up with a 2-colouring decision procedure (colour parent nodes alternate colour from child nodes in a DFS, if a coloured node is encountered: if it is same colour as parent - the graph is not 2-colourable - if it is different - continue colouring).