UG Exam Performance Feedback
Third Year
2019/2020 Semester 1

COMP36111 Advanced Algorithms 1 — Ian Pratt-Hartmann

Comments

1. Candidates essentially demolished this question, displaying a thorough knowledge of Union-Find and its complexity. Part 1a contained a bear trap which many walked into: the question asked for the definition of "non-elementary", and not "non-primitive recursive". (The Ackermann function whose inverse is featured in the last part happens to be both.) Nevertheless, many Candidates correctly answered this (either using the loop-language or in terms of towers of exponentials). Marking was reasonably lenient: candidates clearly displaying the required knowledge but making small slips of expression were given full marks.

2. This question was also very well done. The Gale-Shapley algorithm seemed to be something of a hit with the class: there were almost no wrong answers, and almost no Candidates, I am happy to say, confused the matching algorithm with Gale-Shapley. I was also surprised at the accuracy with which the definition of perfect matching was reproduced. A few marks were lost due to muddled explanations of flow networks and flows. Again, however, Candidates who clearly indicated that thet knew the material were given full marks or nearly full marks (depending, of course, on the level of muddle).

3. Any Candidate who started his answer by writing "NTime(F) means that ..." (and many did) was going to be in for a rocky ride on this question. Despite its straightforward nature (almost no problem-solving involved), it proved to be something of a catastrophe, with large swathes of the class unable to reproduce basic definitions of complexity classes or to give simple proofs of easy relationships between them. Answers were also marred by poor examination technique. When asked to explain what a "Configuration Graph" (for Turing machines) is, simply stating that it is a "graph of configurations", without saying what a configuration is, is not terribly bright. Also, when asked to define PTime and ExpTime,
it is no good to say that the latter is the class of languages recognized by deterministic Turing machines "in exponential time", when we clearly need to know what "exponential time" is. In particular, it's impotant that it is Time(2^{p(n)}) (for p a polynomial), not Time(2^n); otherwise, the the proof in the last part doesn't work. Incidentally, as I expected, only a few Candidates got that proof correct (it's simple if you use the right trick), but I was pleased to see that a few brave Candidates got there in the end.

Generally speaking, on average, the performance was fine.