1999

p5q11
ACN
p12q12

~~6.1 Marking notes~~

~~Powers of two. In binary the only way that the highest 1 bit in a number can not also be present in a-1 is if it is the ONLY bit there.~~

7 Complexity

Explain briefly, stating but not proving any relevant results, which of the following statements are true, which are false and which are meaningless in the context of a study of the complexity of computation: *[each part will be allocated the same weight when marking, but conciseness and clarity of explanation will be important as well as simple factual correctness]*

1. I can check an integer N to see if it is prime by doing test division by all the numbers less than it. This involves just under N trial divisions, and division has a polynomial cost. Therefore testing to see if a number is prime is a problem in the class P;

2. If I am given an integer N and am told in advance that it is composite then I can guess a pair of integers P and Q, multiply them together and check if their product is N. Multiplication has polynomial cost hence factorising known-composites is in the class NP;

3. The quotient of a pair on n-digit integers can be computed in a time less than $kn^{1.1}$ for some value k which depends on the exact speed of the (ordinary) computer being used;

4. If P is a class of problems, and every instance of P can be converted (efficiently) into an instance of a an NP-complete problem Q and a solution to the corresponding instance of Q lets you (again efficiently) derive a solution to the original instance of P, then P is NP-complete;

5. If P = NP then we can solve the decision version of the Travelling Salesman Problem efficiently on a deterministic computer: ie given a graph with weighted edges and an integer k we can find a route visiting each vertex of the graph and having total edge-weight at most k. Because of this we could then solve the minimisation version of the same problem, ie find the shorted path through the graph that visits each vertex, and this would still be achievable in polynomial time.

7.1 Marking notes

1. I can check an integer N to see if it is prime by doing test division by all the numbers less than it. This involves just under N trial divisions, and division has a polynomial cost. Therefore testing to see if a number is prime is a problem in the class P;

p5 q 11
ACN

Well the size of the integer here should be recorded as $\log(N)$ so this bit is wrong.

2. If I am given an integer N and am told in advance that it is composite then I can guess a pair of integers P and Q, multiply them together and check if their product is N. Multiplication has polynomial cost hence factorising known-composites is in the class NP;

   This one however is true. But note we have NOT said it is NP-complete.

3. The quotient of a pair on n-digit integers can be computed in a time less than $kn^{1.1}$ for some value k which depends on the exact speed of the (ordinary) computer being used;

   Yes, it can be done in around $O(n \log n)$ which is certainly $O(n^{1.1})$. This of course is the other bit of the course!

4. If P is a class of problems, and every instance of P can be converted (efficiently) into an instance of a an NP-complete problem Q and a solution to the corresponding instance of Q lets you (again efficiently) derive a solution to the original instance of P, then P is NP-complete;

   This deduction is back to front!

5. If P = NP then we can solve the decision version of the Travelling Salesman Problem efficiently on a deterministic computer: ie given a graph with weighted edges and an integer k we can find a route visiting each vertex of the graph and having total edge-weight at most k. Because of this we could then solve the minimisation version of the same problem, ie find the shorted path through the graph that visits each vertex, and this would still be achievable in polynomial time.

   Yes, by binary chop. But this related to P=NP means that co-NP also =P.

8 Complexity

Explain what is meant by a deterministic and a non-deterministic Turing Machine and the idea of such machines solving a decision problem.

[7 points]

If a non-deterministic Turing Machine solves a certain problem in at most N time-steps, what information must be noted to document the exact state of the machine at each stage as it performs the calculation?

[5 points]

Part of the information you have just identified will be the sequence of states $q_0, q_1, \ldots$ that the machine goes through. Taking account of the fact