EXPTime algorithms

Is there any problem such that its input size is N; its output size is polynomial in terms of N, yet its running time is super-polynomial(N) on a deterministic Turing machine?

Is there any problem such that its input size is N; its output size is polynomial in terms of N, yet its running time is super-polynomial(N) on a non-deterministic Turing machine?

Thanks!

EDIT:

The algorithm can use more than polynomial(N) space during computation.

The "polynomial(N) output size" was to avoid algorithms who generated output of size exp(N) and thus tribially taking atleast exp(N) time to run.

computational-complexity

edited Jul 14 '10 at 18:14

asked Jul 14 '10 at 5:18

LowerBounds
277 5 18

1 Recall that it's still open whether P is the same as PSPACE (problems which can be solved using polynomial amounts of memory but arbitrary time). So you'd need to find something that provably required super-polynomial amounts of space but didn't require super-polynomial amounts of space to give the output. – Noah Snyder Jul 14 '10 at 5:39

Can you explain more? I think you're saying something deep, but I'm too stupid to understand. – LowerBounds Jul 14 '10 at 5:46

3 Do you need just EXPTime-Complete problem? If so there is one in wikipedia: en.wikipedia.org/wiki/EXPTIME. Given a (determinister for first question, nondeterministic for the second) Turing machine and time T decide wheter it halts after T steps. – falagar Jul 14 '10 at 6:10

1 You say that the output size is polynomial in N, but is this really what you mean? After all, any decision problem has yes/no output, which is constant size. So any decidable set that is not in P would be an example for your first question, and these are abundant. Similarly, any decidable set not in NP would be an example for your second. But perhaps you meant that the problem used only polynomial space altogether (putting it in PSPACE), rather than merely polynomial size output. – Joel David Hamkins Jul 14 '10 at 10:30

3 Answers

Short answer: Yes and yes. For the first question, you could take any $EXPTIME$-complete problem. For the second you could take any $NEXPTIME$-complete problem.

Long answer:

Your first question is answered by the problem:

Given a deterministic Turing machine M, string x, and integer k in binary, does M accept x within k steps?

The output is one bit (yes or no). The above problem is $EXPTIME$-complete, hence it requires time that is exponential in the lengths of M, x, and k. It is crucial that k be written in binary. If it were written in unary (as a string of k ones) then it is solvable in polynomial time by direct simulation.

Your second question is answered by the problem:

Given a nondeterministic Turing machine N, string x, and integer k in binary, is there an accepting computation path in N(x) that has at most k steps?

Again the output is just one bit. The above problem is $NEXPTIME$-complete, and hence requires exponential time even on a nondeterministic machine. Again it is crucial that k is written in binary; if it were written in unary then the above problem is NP-complete, and is more commonly known as the "Bounded Halting Problem".

This can all be found in the early chapters of any text on complexity theory.

answered Jul 14 '10 at 6:40

Ryan Williams
3,854 16 33

Oops... didn't realize that part of my answer was just repeating what others wrote (I just quickly typed in an answer and went on my way). Hope it helps, nevertheless – Ryan Williams Jul 14 '10 at 7:28

@Ryan: basically because solving it any "faster" than simulating k steps, in the general case would be akin to solving the halting problem? – LowerBounds Jul 14 '10 at 18:15

@unknown: Yes, it is akin to solving the halting problem, in that from both assumptions you can derive a contradiction via diagonalization. The same proof strategy goes through with little modification. – Ryan Williams Jul 14 '10 at 21:02

can someone elaborate on what the computation path means? like the decision the algorithm makes at each point of branching? – Pita Sep 3 '15 at 5:39

Remember that by the time hierarchy theorem we are assured that there is some decision problem (output size = 1) in EXP that is not in P. So the trivial answer to your question is YES. Of course, if you want a specific problem, then something EXP-complete as @falagar mentions will do it. if you want instead a 'natural problem', then I'm not sure what the answer is.

answered Jul 14 '10 at 6:20

Suresh Venkat
3,337 15 29

A number of EXPTIME-complete problems are listed here.

They include some interesting ones about games, such as generalized chess, checkers, and Go.

answered Jul 14 '10 at 10:32

John Stillwell
9,609 11 70 103
