

Re: interview question on primes

Source: <http://sci.tech-archive.net/Archive/sci.math/2008-02/msg04296.html>

- *From:* "Mark Nudelman" <markn@xxxxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Tue, 26 Feb 2008 14:29:45 -0600
-

quasi wrote:

On Mon, 25 Feb 2008 20:13:45 -0800, "1787" <nobody@xxxxxxxx> wrote:

"quasi" <quasi@xxxxxxxx> wrote in message
news:7ar6s39mvi8cgr09h9umkhjb1r7gjp419@xxxxxxxx

On Tue, 26 Feb 2008 12:28:27 +1100, "Peter Webb"
<webbfamily@xxxxxxxxxxxxxxxxxxxxxxxx> wrote:

"Digital Puer"
<digital_puer@xxxxxxxx> wrote in
message
news:3bdc49e5-d202-491a-bde2-c7c31bee801f@xxxxxxxxxxxxxxxxxxxxxxxx
s.com...

Got this on a software
engineering interview
question:
given two integers A and B,
find all primes between
them.

I basically wrote an initial
function, bool isPrime(int i),
which
loops over all numbers
between 2 and sqrt(i) to see
if i%num ==

0, >>>> in which case the number is not a prime. With this function,
I then >>>> loop between A and B, calling isPrime() on each value.

Re: interview question on primes

Any better ideas?

The "correct" answer is heavily dependent on the range of values of A and B, and if this is all the question states, its a pretty poor question.

No, not really.

It's a real world question, not an academic one.

Moreover, the question was not a "take-home project". Presumably it was expected to be solved right then and there, at the interview.

Judgement is required here.

Of course, the applicant can always ask the interviewer about the range of values, but in my opinion, having to ask that question

would >> show a lack of common sense. The very fact that the range was not >> specified, together with the implied time constraints of the >> interview, should be enough for the applicant to opt for a clean, >> simple, standard solution.

quasi

The OP was asked to find the primes between A and B. His answer was essentially to write the simplest of all programs to determine primality and test all the numbers between A and B. This is about as far from a "real world" answer as you can get. He was obligated to ask questions to more accurately determine the scope of the problem. The useful answer then would be to suggest algorithms that will actually find the primes for values of A and B appropriate to their absolute and relative magnitudes. Peter Webb (first responder to the OP's question) gets the job.

No he doesn't -- not if I'm hiring.

The fact is, most programmers in the world are bad ones.

Separating good from bad at interview time is not so easy, but a

Re: interview question on primes

simple challenger exercise, at the level of difficulty of the one being discussed, is not so bad as a quick test, at least to filter out

- (1) those who are incompetent mathematically
- (2) those who have no design skills
- (3) those who can't program their way out of a paper bag
- (4) those who write overly complex code when the context of the situation (interview time, on the spot) calls for simplicity.

quasi

Indeed. There is a very good argument made in this article: <http://tinyurl.com/yerwac> for giving an interview candidate some very simple programming tasks during the interview. The goal isn't to see how the person accomplishes any real-life tasks (which would be impossible to gauge in an interview timeframe), but just to see if he's fluent in the very basics of programming. Can he write a `strlen` function? Can he write a function to tell whether a string is all caps? Etc. The thing to watch for is how **quickly** he writes the code. A superstar will be able to write such code about as fast as he can write. An inferior programmer will probably be able to write the code, but it will take much longer.

--Mark

.