

# Re: NP-hard problems

---

*Source:* <http://sci.tech-archive.net/Archive/sci.math/2008-03/msg03803.html>

---

- *From:* quasi <quasi@xxxxxxxx>
  - *Date:* Thu, 27 Mar 2008 12:23:50 -0500
- 

On Thu, 27 Mar 2008 15:58:54 +0100, saneman <asdfsdf@xxxxxxxx> wrote:

Timothy Murphy wrote:

saneman wrote:

As I understand a problem is said to be NP-Hard if there is no polynomial time algorithm that can solve it.

But what if I have a logarithmic algorithm that solves the problem?  
Logarithmic functions are not polynomial.

I wonder if you really have a "logarithmic algorithm"?  
I imagine they are quite rare.

Comparing the first 2 elements is  $O(\log n)$ .

Remember that it is polynomial time in the `_length_` of the input, so if you are talking about integer input, it is polynomial time in  $\log n$ .

Its possible to find an element in  $\lg n$  time (asymptotic notation) in a red/black tree.

I don't think so.

A single comparison is  $O(\log n)$ , but you need  $O(\log n)$  such comparisons, hence the actual search time is  $O((\log n)^2)$ , not  $O(\log n)$ .

quasi

.