CS 173, Fall 2014
Examlet 11, Part B

NETID:

FIRST: | **LAST:**

Discussion: Thursday 2 3 4 5 Friday 9 10 11 12 1 2

(15 points) Check the (single) box that best characterizes each item.

The running time of the Towers of Hanoi solver

$\Theta(\log n)$ ☐ $\Theta(n \log n)$ ☐ $\Theta(n^2)$ ☐ $\Theta(2^n)$ ☑

The running time of Karatsuba's algorithm is recursively defined by $T(1) = d$ and $T(n) =$

$2T(n/2) + cn$ ☐ $3T(n/2) + cn$ ☑

$4T(n/2) + cn$ ☐ $4T(n/2) + c$ ☐

Finding the chromatic number of a graph with n nodes requires $\Theta(2^n)$ steps.

proven true ☐ proven false ☐

not known ☑

Algorithm A takes n^5 time and Algorithm B takes 2^n time. On one input, A takes x time and B takes y time. How long will each algorithm take if I double the input size?

A: $2x$, B: $2y$ ☐ A: $5x$, B: $2y$ ☐ A: $32x$, B: $2y$ ☐

A: $32x$, B: y^2 ☑ A: $32x$, B: 2^y ☐ A: x^5, B: 2^y ☐

If a yes/no problem is in NP, a "yes" answer always has a succinct justification.

true ☑ false ☐ not known ☐

CS 173, Fall 2014
Examlet 11, Part B

NETID:

FIRST: | **LAST:**

Discussion: Thursday 2 3 4 5 Friday 9 10 11 12 1 2

(15 points) Check the (single) box that best characterizes each item.

The running time of the Towers of Hanoi solver is recursively defined by $T(1) = d$ and $T(n) =$

- $2T(n-1)+c$ [√]
- $2T(n-1)+cn$ [ ]
- $2T(n/2)+c$ [ ]
- $2T(n/2)+cn$ [ ]

Problems in NP need exponential time

- proven true [ ]
- proven false [ ]
- not known [√]

The running time of mergesort is $O(n^3)$.

- True [√]
- False [ ]

$n^{log_2 3}$ grows

- faster than n^2 [ ]
- slower than n^2 [√]
- at the same rate as n^2 [ ]

It takes exponential time to determine whether a propositional logic expression can be made true by picking the right true/false values for its propositional variables (e.g. p, q, r).

- proven true [ ]
- proven false [ ]
- not known [√]

CS 173, Fall 2014
Examlet 11, Part B

NETID:

FIRST: **LAST:**

Discussion: **Thursday** 2 3 4 5 **Friday** 9 10 11 12 1 2

(15 points) Check the (single) box that best characterizes each item.

The solution to the Tower of Hanoi puzzle with n disks requires $\Theta(2^n)$ steps

proven true [√] proven false [ ] not known [ ]

The running time of mergesort

$\Theta(n)$ [ ] $\Theta(n \log n)$ [√]

$\Theta(n^2)$ [ ] $\Theta(2^n)$ [ ]

Problems in class P (as in P vs. NP) require exponential time

never [√] sometimes [ ]

always [ ] not known [ ]

$T(1) = d$
$T(n) = 3T(n/2) + n$

$\Theta(n)$ [ ] $\Theta(n \log n)$ [ ] $\Theta(n^2)$ [ ]

$\Theta(n^{\log_3 2})$ [ ] $\Theta(n^{\log_2 3})$ [√] $\Theta(2^n)$ [ ]

The Marker Making problem can be solved in polynomial time.

proven true [ ] proven false [ ] not known [√]

CS 173, Fall 2014
Examlet 11, Part B

NETID:

FIRST: | **LAST:**

Discussion: **Thursday** **2** **3** **4** **5** **Friday** **9** **10** **11** **12** **1** **2**

(6 points) Fill in the missing bits of the recursive algorithm for solving the Towers of Hanoi puzzle.

hanoi(A,B,C: pegs, $d_1, d_2 \ldots d_n$: disks) \\ move n disks from peg A to peg B

if $(n = 1)$ move d_1 from A to B

else

hanoi(A,C,B,$d_1, \ldots, d_{n-1}$)

move d_n from A to B

hanoi(C,B,A,$d_1, \ldots, d_{n-1}$)

(9 points) Check the (single) box that best characterizes each item.

Problems in class NP (as in P vs. NP) can be solved in exponential time

never ☐ sometimes ☐

always ☑ not known ☐

Karatsuba's integer multiplication algorithm

$\Theta(n^2)$ ☐ $\Theta(n^3)$ ☐ $\Theta(n \log n)$ ☐

$\Theta(n^{log_2 3})$ ☑ $\Theta(n^{log_3 2})$ ☐ $\Theta(2^n)$ ☐

Merging two sorted lists of numbers

$O(\log n)$ ☐ $O(n)$ ☑ $O(n \log n)$ ☐ $O(n^2)$ ☐