CS 173, Spring 2015
Examlet 11, Part B

NETID:

FIRST: | **LAST:**

Discussion: Monday 9 10 11 12 1 2 3 4 5

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

Karatsuba's integer multiplication algorithm

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

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

Determining whether a graph with n edges is connected.

polynomial ☑ exponential ☐ in NP ☐

The running time of the Towers of Hanoi solver

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

Algorithm A takes 2^n time. On one input, A takes x time. How long will it take if I double the input size?

$2x$ ☐ 2^x ☐ x^2 ☑

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

True ☑ False ☐

CS 173, Spring 2015
Examlet 11, Part B

NETID:

FIRST:	**LAST:**

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

(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. On one input, A takes x time. How long will it take if I double the input size?

$2x$ ☐ $5x$ ☐ $32x$ ☑ x^5 ☐

Producing all parses for a sentence.

polynomial ☐ exponential ☑ in NP ☐

CS 173, Spring 2015
Examlet 11, Part B

NETID:

FIRST: | **LAST:**

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

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

The running time of binary search is recursively defined by $T(1) = d$ and $T(n) =$

- $T(n/2) + c$ ☑ √
- $T(n/2) + cn$ ☐
- $2T(n/2) + c$ ☐
- $2T(n/2) + cn$ ☐

Problems in NP need exponential time

proven true ☐ proven false ☐ not known ☑ √

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

True ☑ √ False ☐

Algorithm A takes 2^n time. On one input, A takes x time. How long will it take if I double the input size?

$2x$ ☐ 2^x ☐ x^2 ☑ √

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

true ☑ √ false ☐ not known ☐

CS 173, Spring 2015
Examlet 11, Part B

NETID:

FIRST: | **LAST:**

Discussion: Monday 9 10 11 12 1 2 3 4 5

(6 points) Fill in the missing bits of this recursive algorithm for returning the location of a number k in a sorted list of numbers $a_p, a_2, ..., a_q$.

search(p,q,k) \\ assume $p \leq q$

$m := \lfloor (p+q)/2 \rfloor$

if $k = a_m$ then return m

else if $(k < a_m)$ and $p < m$ then

Solution: search(p,m-1,k)

else if $(k > a_m)$ and $q > m$ then

Solution: search(m+1,q,k)

else return -1 \\ i.e. error, not found

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

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

CS 173, Spring 2015
Examlet 11, Part B

NETID:

FIRST: **LAST:**

Discussion: Monday 9 10 11 12 1 2 3 4 5

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

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

never [√] sometimes [ ]

always [ ] not known [ ]

The running time of binary search

$\Theta(\log n)$ [√] $\Theta(n)$ [ ]

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

The Marker Making problem can be solved in polynomial time.

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

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

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

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

Producing all parses for a sentence.

polynomial [ ] exponential [√] in NP [ ]

CS 173, Spring 2015
Examlet 11, Part B

NETID:

FIRST:	**LAST:**

Discussion: Monday 9 10 11 12 1 2 3 4 5

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

Karatsuba's integer multiplication algorithm

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

The Travelling Salesman problem can be solved in polynomial time.

proven true ☐ proven false ☐ not known ☑ $\surd$

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

$2T(n/2) + cn$ ☐	$3T(n/2) + cn$ ☑ $\surd$
$4T(n/2) + cn$ ☐	$4T(n/2) + c$ ☐

$n^{log_2 3}$ grows

faster than n^2 ☐ slower than n^2 ☑ $\surd$ at the same rate as n^2 ☐

Producing all parses for a sentence.

polynomial ☐ exponential ☑ $\surd$ in NP ☐