CS 173, Spring 19 Examlet 11, white sheet

Name:____

NetID:______________________ **Lecture:** **A** **B**

Discussion: **Thursday** **Friday** **9** **10** **11** **12** **1** **2** **3** **4** **5** **6**

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

Algorithm A takes $\log_2 n$ time. On one input, A takes x time. How long will it take if I double the input size?

$x+1$ [√] $2x$ [ ] 2^x [ ] x^2 [ ]

$T(1) = c$
$T(n) = 4T(n/2) + n$

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

The running time of binary search is $O(n \log n)$.

true [√] false [ ]

For a problem to satisfy the definition of NP, a "yes" answer must have a succinct justification.

true [√] false [ ]

Deciding whether an input logic expression be made true by appropriate choice of input values.

polynomial [ ] exponential [ ] in NP [√]

Name:____

NetID:________________________ **Lecture: A B**

Discussion: Thursday Friday 9 10 11 12 1 2 3 4 5 6

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

Karatsuba's integer multiplication algorithm

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

$T(1) = d$
$T(n) = 2T(n/4) + n$

$\Theta(\log n)$ [ ]	$\Theta(\sqrt{n})$ [ ]	$\Theta(n)$ [✓]	$\Theta(n \log n)$ [ ]
$\Theta(n^2)$ [ ]	$\Theta(n^3)$ [ ]	$\Theta(2^n)$ [ ]	$\Theta(3^n)$ [ ]

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

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

Circuit satisfiability can be solved in polynomial time.

true [ ] false [ ] not known [✓]

For a problem to satisfy the definition of co-NP, a "no" answer must have a succinct justification.

true [✓] false [ ]

CS 173, Spring 19 — Examlet 11, white sheet

Name:____________________________________

NetID:____________________ **Lecture:** **A** **B**

Discussion: **Thursday** **Friday** **9** **10** **11** **12** **1** **2** **3** **4** **5** **6**

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

The running time of merge

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

$T(1) = d$
$T(n) = T(n-1) + n$

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

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

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

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

true ☑ √ false ☐ not known ☐

The chromatic number of a graph with n nodes can be found in polynomial time.

true ☐ false ☐ not known ☑ √

Name:______________________________________

NetID:____________________ **Lecture:** **A** **B**

Discussion: **Thursday** **Friday** **9** **10** **11** **12** **1** **2** **3** **4** **5** **6**

(6 points) Fill in the missing bits of a recursive implementation of Merge, which merges two lists of integers sorted in increasing order. Use the functions first (first element), rest (everything after first element), and and cons (adds number to list).

Merge(L_1,L_2: sorted lists of real numbers)

if (L_1 is empty and L_2 is empty) return emptylist

else if (L_2 is empty or first(L_1) <= first(L_2))

Solution: return cons(first(L_1),merge(rest(L_1),L_2))

else

Solution: return cons(first(L_2),merge(L_1,rest(L_2)))

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

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

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

The Towers of Hanoi puzzle requires exponential time.

true ☑ false ☐ not known ☐

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

true ☐ false ☐ not known ☑

Name:____

NetID:____________________ **Lecture:** **A** **B**

Discussion: **Thursday** **Friday** **9** **10** **11** **12** **1** **2** **3** **4** **5** **6**

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

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

$4T(n/2) + cn$	☐	$4T(n/2) + c$	☐
$2T(n/2) + cn$	☐	$3T(n/2) + cn$	√

$T(1) = d$
$T(n) = 2T(n-1) + c$

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

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

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

For a problem to satisfy the definition of co-NP, a "yes" answer must have a succinct justification. true ☐ false √

The Towers of Hanoi puzzle can be solved in polynomial time. true ☐ false √ not known ☐

Name:____

NetID:____________________ **Lecture: A B**

Discussion: Thursday Friday 9 10 11 12 1 2 3 4 5 6

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

Adding element to start of array (array gets longer)

$\Theta(1)$	☐	$\Theta(\log n)$	☐	$\Theta(n)$	☑	$\Theta(n \log n)$	☐
$\Theta(n^2)$	☐	$\Theta(n^3)$	☐	$\Theta(2^n)$	☐	$\Theta(3^n)$	☐

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

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

$T(1) = c$
$T(n) = 2T(n/2) + n^2$

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

Problems in class NP require exponential time

true ☐ false ☐ not known ☑

The Marker Making problem can be solved in polynomial time.

true ☐ false ☐ not known ☑

Name:____

NetID:________________________ **Lecture:** **A** **B**

Discussion: **Thursday** **Friday** **9** **10** **11** **12** **1** **2** **3** **4** **5** **6**

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

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

$\Theta(\log n)$	☑	$\Theta(\sqrt{n})$	☐	$\Theta(n)$	☐	$\Theta(n \log n)$	☐
$\Theta(n^2)$	☐	$\Theta(n^3)$	☐	$\Theta(2^n)$	☐	$\Theta(3^n)$	☐

Dividing a linked list in half

$\Theta(1)$	☐	$\Theta(\log n)$	☐	$\Theta(n)$	☑	$\Theta(n \log n)$	☐
$\Theta(n^2)$	☐	$\Theta(n^3)$	☐	$\Theta(2^n)$	☐	$\Theta(3^n)$	☐

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

$2T(n/2) + c$	☐	$2T(n/2) + cn$	☐
$2T(n-1) + c$	☑	$2T(n-1) + cn$	☐

Producing all parses for a sentence.

polynomial	☐	exponential	☑	in NP	☐

The Travelling Salesman Problem

polynomial	☐	exponential	☐	in NP	☑

CS 173, Spring 19 **Examlet 11, white sheet**

Name:____

NetID:____________________ **Lecture:** **A** **B**

Discussion: **Thursday** **Friday** **9** **10** **11** **12** **1** **2** **3** **4** **5** **6**

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

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

$\Theta(\log n)$	☐	$\Theta(\sqrt{n})$	☐	$\Theta(n)$	☑ √	$\Theta(n \log n)$	☐
$\Theta(n^2)$	☐	$\Theta(n^3)$	☐	$\Theta(2^n)$	☐	$\Theta(3^n)$	☐

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

$x + 2$ ☐ $2x$ ☑ √ 2^x ☐ x^2 ☐

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

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

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

true ☐ false ☑ √ not known ☐

The Travelling Salesman problem can be solved in polynomial time.

true ☐ false ☐ not known ☑ √