CS 173, Fall 2015
Examlet 11, Part B

NETID:

FIRST: | **LAST:**

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

(6 points) Your partner has implemented the function Merge(A,B), which merges two sorted linked lists of integers. Using Merge, fill in the missing parts of this implementation of Mergesort.

Mergesort($L = (a_1, a_2, \ldots, a_n)$) \\ input is a linked list L containing n integers

Solution: if (n=1) return L

p = floor(n/2)

Solution:

$L_a = (a_1, \ldots, a_p)$
$L_b = (a_{p+1}, \ldots, a_n)$
return Merge(Mergesort(L_a), Mergesort(L_b))

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

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

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

The Towers of Hanoi puzzle can be solved in polynomial time.

proven true ☐ proven false ☑ not known ☐

Merging two sorted lists

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

CS 173, Fall 2015
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.

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

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

Circuit satisfiability can be solved in polynomial time.

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

The running time of the Towers of Hanoi solver

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

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

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

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

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

$4T(n/2) + cn$ ☐ $4T(n/2) + c$ ☐

CS 173, Fall 2015
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.

Karatsuba's integer multiplication algorithm

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

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

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

$T(n/2) + c$ ☑ $T(n/2) + cn$ ☐

$2T(n/2) + c$ ☐ $2T(n/2) + cn$ ☐

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

true ☑ 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 ☐

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

never ☑ sometimes ☐

always ☐ not known ☐

CS 173, Fall 2015
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 $T(n) =$

$2T(n-1)+c$ ☑ $2T(n-1)+cn$ ☐

$2T(n/2)+c$ ☐ $2T(n/2)+cn$ ☐

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

true ☑ false ☐ not known ☐

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 ☑

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

proven true ☐ proven false ☐

not known ☑

CS 173, Fall 2015
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 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.

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 ☑

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, Fall 2015
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, $d_1, d_2 \dots d_n$: disks) \\\\ move n disks from peg A to peg B

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

else

Solution:
hanoi(A,C,B: pegs, $d_1, d_2 \dots d_{n-1}$: disks) \\\\ move smaller disks to C

move d_n from A to B

Solution:
hanoi(C,B,A: pegs, $d_1, d_2 \dots d_{n-1}$: disks) \\\\ move smaller disks to B

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

Determining whether a graph with n edges is connected.

polynomial ☑ exponential ☐ in NP ☐

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

$2T(n-1) + c$ ☐ $2T(n-1) + cn$ ☐

$2T(n/2) + c$ ☐ $2T(n/2) + cn$ ☑

The running time of binary search

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