Started on	Tuesday, 16 November 2021, 10:15 AM
State	Finished
Completed on	Tuesday, 16 November 2021, 10:40 AM
Time taken	24 mins 11 secs
Grade	**40.0** out of 48.0 (**83**%)

Question 1

Complete

Mark 4.0 out of 4.0

Flag question

“There exists a decidable function that is not time constructible”. Which of the following statements is true about this statement:

Select one:

a.
The above statement is false as all functions are not decidable.

b.
The above statement is false as all decidable functions are time constructible.

c.
The above statement can be proven to be true by the fact that $\neg HP$ is not decidable, and therefore not time constructible.

d.
The above statement can be proven to be true by a diagonalization argument on decidable functions.

The correct answer is:
The above statement can be proven to be true by a diagonalization argument on decidable functions.

Question 2

Complete

Mark 4.0 out of 4.0

Flag question

For every i, $\Sigma_i^p = \sum_c(\Sigma_i Time(n^c))$. Which of the following statements is true:

Select one:

a.

This statement is false and it is only true that $\sum_c(\Sigma_i Time(n^c)) \subsetneq \Sigma_i^p$.

b.
This statement is false and it is only true that $\Sigma_i^p \subsetneq \sum_c(\Sigma_i Time(n^c))$.

c.
This statement is true and by a similar argument it can be shown that $\Pi_i^p = \sum_c(\Pi_i Time(n^c))$.

d.
This statement is true but there is an i such that $\Pi_i^p \subsetneq \sum_c(\Pi_i Time(n^c))$ because the first quantifier is a $\forall$ quantifier.

The correct answer is:
This statement is true and by a similar argument it can be shown that $\Pi_i^p = \sum_c(\Pi_i Time(n^c))$.

Question 3

Complete

Mark 4.0 out of 4.0

Flag question

Let CL_1 and CL_2 be two time complexity classes or two space complexity classes. Let $c \geq 1$ be a positive integer.

Select one:

a.
If $CL_1(f(n)) \subsetneq CL_2(g(n))$, then $CL_1(f(n^c)) \subsetneq CL_2(g(n^c))$ only when they are two time complexity classes.

b.
If $CL_1(f(n))[\backslash subsetneq?]CL_2(g(n))$, then $CL_1(f(n^c))[\backslash subsetneq?]CL_2(g(n^c))$ only when they are two space complexity classes.

c.
If $CL_1(f(n))[\backslash subsetneq?]CL_2(g(n))$, then $CL_1(f(n^c))[\backslash subsetneq?]CL_2(g(n^c))$

no matter if they are two time complexity classes or two space complexity classes.

d.
If $CL_1(f(n))[\subsetneq?]CL_2(g(n))$, then there is always a c such that $CL_1(f(n^c)) = CL_2(g(n^c))$.

The correct answer is:
If $CL_1(f(n))[\subsetneq?]CL_2(g(n))$, then $CL_1(f(n^c))[\subsetneq?]CL_2(g(n^c))$ no matter if they are two time complexity classes or two space complexity classes.

Question 4

Complete

Mark 0.0 out of 4.0

Flag question

The union of 2 NL-complete languages is NL-complete.

Select one:

a.
True. The composition of the reduction functions is still a reduction function.

b.
True. By Savitch's theorem, we can show that there exists a reduction function for the union language.

c.
False. This cannot be true as a co-NL complete language is also NL-complete and this would mean L = NL.

d.
False. This cannot be true as the class NL itself is not closed under union.

The correct answer is:
False. This cannot be true as a co-NL complete language is also NL-complete and this would mean L = NL.

Question 5

Complete

Mark 4.0 out of 4.0

Flag question

The language BIPARTITE consists of graph encodings where the graph is bipartite. Which of the following is true about this language?

Select one:

a.
BIPARTITE is not in NL because it is not in P itself.

b.
BIPARTITE is in P. But it is not in NL as otherwise this would imply L = NL, and that would mean P = NP.

c.
BIPARTITE is in NL because an $O(n^c)$ time algorithm means that it can take at most $\log(n^c) = c\log(n)$ space by a method of binary coding.

d.
BIPARTITE is in NL, as ¬BIPARTITE can easily be shown to be in NL and then we can apply Immerman-Szelepzcenyi Theorem.

The correct answer is:
BIPARTITE is in NL, as ¬BIPARTITE can easily be shown to be in NL and then we can apply Immerman-Szelepzcenyi Theorem.

Question 6

Complete

Mark 4.0 out of 4.0

Flag question

For a given positive integer $c \geq 1$, which of the following can be proven?

Select one:

a.
$P \subseteq DSPACE((\log n)^c) \subseteq NP \subseteq NSPACE((\log n)^c)$

b.
$DSPACE((\log n)^c) \subseteq P \subseteq NSPACE((\log n)^c) \subseteq NP$

c.
$DSPACE((\log n)^c) \subseteq NSPACE((\log n)^c) \subseteq P \subseteq NP$

d.
$DSPACE((\log n)^c) = NSPACE((\log n)^c) \subseteq P \subseteq NP$

The correct answer is:
$DSPACE((\log n)^c) \subseteq NSPACE((\log n)^c) \subseteq P \subseteq NP$

Question 7

Complete

Mark 4.0 out of 4.0

Flag question

The language TRIANGLE consists of graph encodings where the graph has a triangle. Which of the following is true?

Select one:

a.
TRIANGLE is not in L but in NL.

b.
TRIANGLE is not in NL.

c.
TRIANGLE is in L.

d.
TRIANGLE is not in P.

The correct answer is:
TRIANGLE is in L.

Question 8

Complete

Mark 0.0 out of 4.0

Flag question

We have a nondeterministic space hierarchy theorem (NSHT), but there will be differences with the Space Hierarchy theorem:

Select one:

a.
The condition $f(n) = o(g(n))$ will not be sufficient to show that $NSPACE(f(n))[\backslash \mathrm{subsetneq?}]NSPACE(g(n))$ for the NSHT.

b.
NSHT can be proven in a very similar way for the condition of $f(n) = o(g(n))$ but by using Savitch's theorem.

c.
NSHT can be proven in a very similar way for the condition of $f(n) = o(g(n))$ but by using Immerman-Szelepzcenyi Theorem.

d.
We do not have a NSHT.

The correct answer is:
NSHT can be proven in a very similar way for the condition of $f(n) = o(g(n))$ but by using Immerman-Szelepzcenyi Theorem.

Question 9

Complete

Mark 4.0 out of 4.0

Flag question

Consider the algorithm Rand-Ter-QS. Here quicksort is done, but at each step we pick 2 pivot elements p_1, p_2. Without loss of generality, $p_1 \leq p_2$. We divide the rest of the elements into 3 parts depending on whether they are $\leq p_1$, between p_1 and p_2, or $\geq p_2$. Then the 3 sets are recursively sorted and finally put together in linear time. What is the expected number of comparisons in Rand-Ter-QS:

Select one:

a. $O(n \log n)$

b. $O(n \log \log n)$

c. $O(n^{3/2} \log n)$

d. $O(n)$

The correct answer is:
$O(n \log n)$

Question 10

Complete

Mark 4.0 out of 4.0

Flag question

Let n be the size of the input of an RP algorithm, and d be a positive integer. The error probability of this RP algorithm can be reduced to e^{-n^d} by

Select one:

a.

Repeating the algorithm independently $O(n^d)$ times, and if there is a YES returned at least once then the output is YES.

b.

Repeating the algorithm independently $O(n^d)$ times and returning the majority of YESes or NOs as the output.

c.
Both strategies work

d.
Neither strategy can be used for the above error reduction.

The correct answer is:
Both strategies work

Question 11

Complete

Mark 4.0 out of 4.0

Flag question

PP is a complexity class where $L \in PP$:

$$x \in L [\iff?] \mathbb{P}[M(x)=1] > 1/2$$

$$x \notin L [\iff?] \mathbb{P}[M(x)=1] < 1/2$$

Select one:

a.
$NP \subseteq PP \subseteq PSPACE$: [Proof sketch] The first containment can be argued once you analyse an algorithm for SAT where a satisfying assignment is picked uniformly at random; if it satisfies the input instance then the output is YES, otherwise with probability $\frac{1}{2}+\frac{1}{2^{n+1}}$ the output is NO, and with probabilty $\frac{1}{2}-\frac{1}{2^{n+1}}$ the output is YES. The second containment can be argued by trying all choices of random sequences.

b.
$NP, PP \subseteq PSPACE$ but NP and PP cannot be compared: [Proof sketch] From defintion $PP = BPP$, and therefore the relation between NP and PP are not known. But both belong to $PSPACE$, as one has to try all possible certificates and random sequences to show containment in $PSPACE$.

c.
$PP \subseteq NP \subseteq PSPACE$: [Proof sketch] The first containment can be argued once you analyse an algorithm in class PP to have at least 1 random sequence that leads to YES for a YES instance. The second containment is by trying all possible random sequences/certificates in polynomial space.

d.
$NP \subseteq PSPACE \subseteq PP$: [Proof sketch] The first containment has been taught in class. The second containment can be argued by analysing the algorithm where a $PSPACE$ algorithm is taken as a subroutine, a coin is tossed polynomially many times and then the random sequences are ignored.

The correct answer is:
$NP \subseteq PP \subseteq PSPACE$: [Proof sketch] The first containment can be argued once you analyse an algorithm for SAT where a satisfying assignment is picked uniformly at random; if it satisfies the input instance then the output is YES, otherwise with probability $\frac{1}{2}+\frac{1}{2^{n+1}}$ the output is NO, and with probabilty $\frac{1}{2}-\frac{1}{2^{n+1}}$ the output is YES. The second containment can be argued by trying all choices of random sequences.

Question 12

Complete

Mark 4.0 out of 4.0

Flag question

Statement I: It is not known whether SAT is in BPP, and this is an open question.

Statement II: If SAT is in BPP, then an RP algorithm for SAT can also be designed by recursively using the BPP algorithm.

Select one:

a. Both I and II are true

b.
Both I and II are false

c. I is true but II is false

d.
I is false but II is true

The correct answer is: Both I and II are true

Finish review

QUIZ NAVIGATION

1 2 3 4 5 6 7 8 9 10 11 12

Finish review