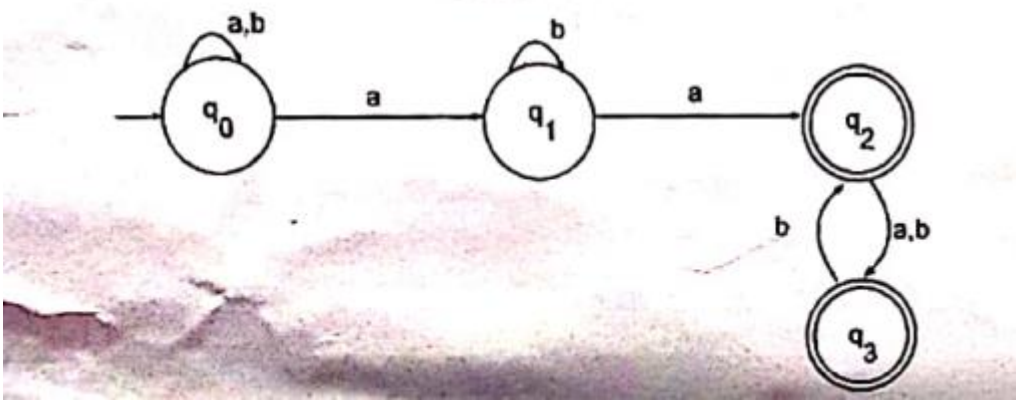
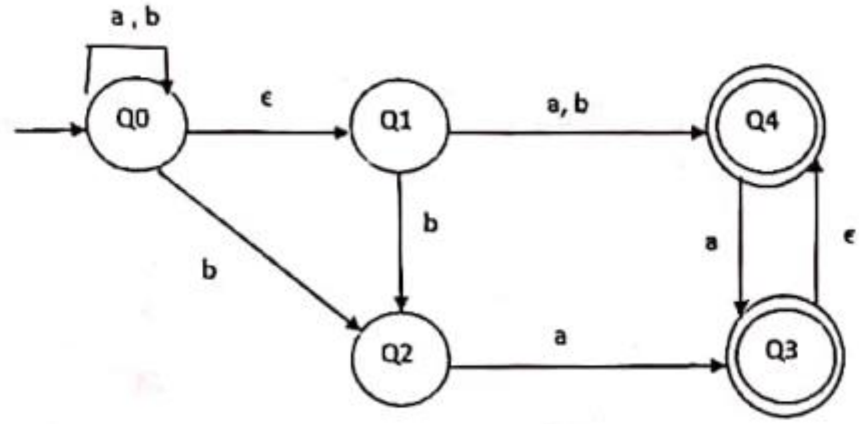
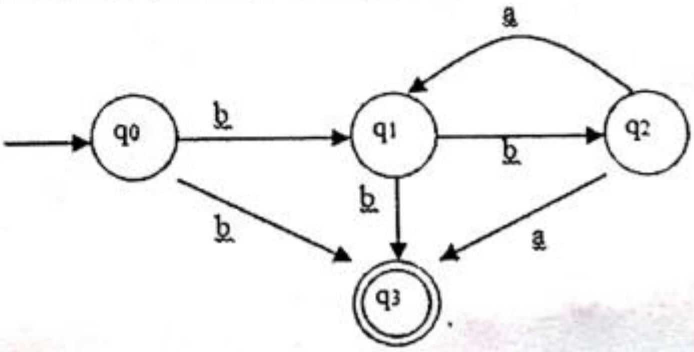
CHAPTER WISE PU BOARD QUESTIONS

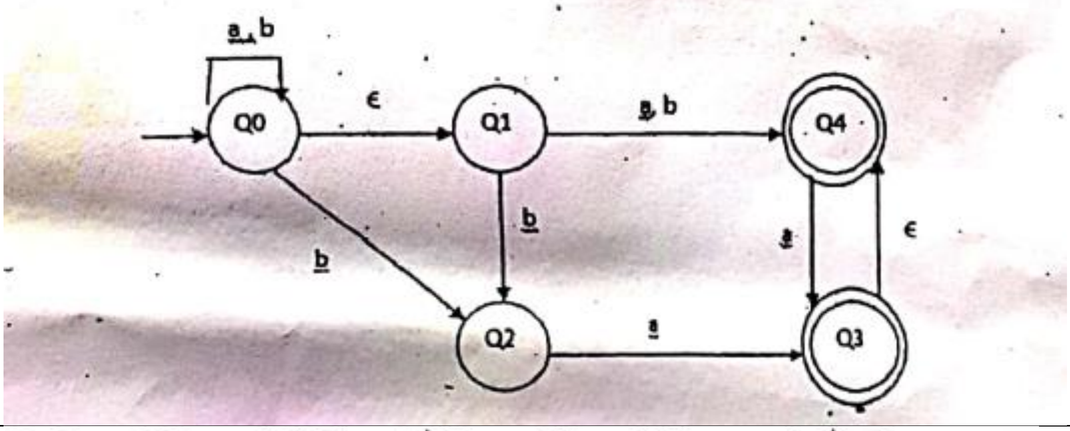
CHAPTER 1: SET THEORY

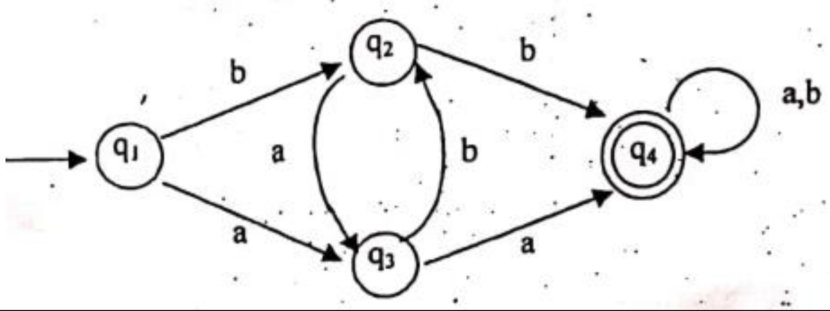
1.	What is set? Show the different types of set operation with examples.	**2020 FALL**
2.	Using De Morgan's Law, prove the following: i) $(A \cup B)' = A' \cap B'$ ii) $(A \cap B)' = A' \cup B'$	**2019 Fall**
3.	Define cartesian product and power set.	**2018 S**
4.	Write short notes on: a) Relation and functions [2020 F, 2019 S, 2018 F, 2017 F,] b) Cartesian product, relation and function [2020 Fall] c) Alphabet and language [2016 S, 2017 S, 2017 S, 2015 S] d) Regular expressions [2019 F] e) Application of RE [2017 S]	

CHAPTER 2: FINITE AUTOMATA (FA)

<table>
<tr><td>1.</td><td>Explain finite automata along with is uses and applications. Construct a DFA that recognizes language L that accepts the set of strings containing exactly four 1's in every string over alphabet $\sum=\{0, 1\}$and test your design with a valid string.</td><td>2020 FALL</td></tr>
<tr><td>2.</td><td>Find the regular expression from NFA M= (K, Σ, Δ, s, F), where K= $\{q_0, q_1, q_2, q_3, q_4, q_5\}$, Σ= {a, b}, s = q_0, F = { q_5} and Δ is given as follows.
<table>
<tr><th>δ / Σ</th><th>a</th><th>b</th><th>∈</th></tr>
<tr><td>→q_0</td><td>-</td><td>-</td><td>q_1</td></tr>
<tr><td>q_1</td><td>q_2</td><td>q_4</td><td>-</td></tr>
<tr><td>q_2</td><td>-</td><td>q_3, q_4</td><td>-</td></tr>
<tr><td>q_3</td><td>q_3</td><td>q_3</td><td>q_5</td></tr>
<tr><td>q_4</td><td>q_2, q_4</td><td>-</td><td>-</td></tr>
<tr><td>*q_5</td><td>-</td><td>-</td><td>-</td></tr>
</table></td><td>2020 FALL</td></tr>
<tr><td>3.</td><td>Convert the following NFA to its equivalent DFA.

</td><td>2019 FALL</td></tr>
<tr><td>4.</td><td>Define pumping Lemma. Show that $L = \{a^n b^{2n} : n \geq 1\}$ is not regular using pumping lemma for regular language.</td><td>2019 FALL</td></tr>
<tr><td>5.</td><td>Define DFA and NFA. Design a FA which accepts the language L={w/w has both and an even number of 0's and an even number of 1's over alphabet $\sum=\{0,1\}$.</td><td>2018 SPRING</td></tr>
<tr><td>6.</td><td>State and explain Arden's theorem with example. Also convert the following regular expresssion to finite automata: 00*(0*0+1)</td><td>2018 SPRING</td></tr>
</table>

<table>
<tr><td>7.</td><td>What is a finite automaton? Design a DFA that accepts the language given by $L=\{w\in\{0,1\}^* : w \text{ does not contain four consecutive 0's}\}$. Hence test your design for 01010001.</td><td>2018 FALL</td></tr>
<tr><td>8.</td><td>Construct a DFA equivalent to NFA as shown:
</td><td>2018 FALL</td></tr>
<tr><td>9.</td><td>Give the formal definition of DFA. Design a FA that accepts a set of string such that every string ends in 00, over alphabet {0,1}.</td><td>2017 FALL</td></tr>
<tr><td>10.</td><td>Construct finite automata for the following regular expression. a(a+b)*bb</td><td>2017 FALL</td></tr>
<tr><td>11.</td><td>Define Finite Automata .Design a FA that accepts set of strings which doesn't starts with 0 and ends with 1 over the given alphabet $\Sigma=\{0,1\}$.</td><td>2017 SPRING</td></tr>
<tr><td>12.</td><td>Minimize the following DFA by using state minimization method. Where ⟶ represents initial state and * represents final state.
<table>
<tr><th>δ / Σ</th><th>0</th><th>1</th></tr>
<tr><td>$\rightarrow q_0$</td><td>q_1</td><td>q_2</td></tr>
<tr><td>$*q_1$</td><td>q_1</td><td>q_3</td></tr>
<tr><td>q_2</td><td>q_2</td><td>q_2</td></tr>
<tr><td>$*q_3$</td><td>q_5</td><td>q_2</td></tr>
<tr><td>$*q_4$</td><td>q_4</td><td>q_2</td></tr>
<tr><td>$*q_5$</td><td>q_4</td><td>q_2</td></tr>
<tr><td>q_6</td><td>q_5</td><td>q_6</td></tr>
<tr><td>q_7</td><td>q_5</td><td>q_6</td></tr>
</table></td><td>2017 SPRING</td></tr>
<tr><td>13.</td><td>Convert the following NFA to its equivalent DFA.
</td><td>2016 FALL</td></tr>
<tr><td>14.</td><td>Sate pumping lemma for regular expression. Mention the closure properties of regular expression.</td><td>2016 FALL</td></tr>
</table>

<table>
<tr><td>15.</td><td>Give the formal definition of DFA. Design a FA that accepts a set of string such that every string ends in 00, over alphabet {0,1}.</td><td>2016 SPRING</td></tr>
<tr><td>16.</td><td>Construct finite automata for the following regular expression.
a(a+b)*bb</td><td>2016 SPRING</td></tr>
<tr><td>17.</td><td>Define Regular Expression. Write regular expression for language L={w ∈ {a , b}*: number of a is divisible by 3}</td><td>2015 FALL</td></tr>
<tr><td>18.</td><td>What is a finite automaton? Design a DFA that accepts the language given by L= {w ∈ {0, 1}*: w has neither '00' nor '11' as substring}. Hence test your design for 01011010.</td><td>2015 FALL</td></tr>
<tr><td>19.</td><td>Construct a DFA equivalent to NFA as shown:

</td><td>2015 FALL</td></tr>
<tr><td>20.</td><td>Define DFA and NFA. Design a FA which accepts the language L={w/w has both an even number of 0's and an even number of 1's over alphabet Σ={0,1}.</td><td>2014 SPRING</td></tr>
<tr><td>21.</td><td>Consider the NFA ({q_0,q_1,q_2}, {a,b,∈}, δ, q_0, {q_1}) with state transition table given as follows:
<table>
<tr><td>δ</td><td>a</td><td>b</td><td>∈</td></tr>
<tr><td>q_0</td><td>{q_1}</td><td>Φ</td><td>Φ</td></tr>
<tr><td>q_1</td><td>{q_1}</td><td>Φ</td><td>{q_2}</td></tr>
<tr><td>q_2</td><td>Φ</td><td>{q_0}</td><td>Φ</td></tr>
</table>
Convert this NFA to its equivalent DFA.</td><td>2014 SPRING</td></tr>
<tr><td>22.</td><td>Construct an NFA for the regular expression 01*+1.</td><td>2014 SPRING</td></tr>
<tr><td>23.</td><td>State and prove the pumping lemma for regular sets.</td><td>2014 SPRING</td></tr>
<tr><td>24.</td><td>Prove that the language L={a^nba^n for n=0,1,2,3,...} is not regular.</td><td>2014 SPRING</td></tr>
</table>

<table>
<tr><td>25.</td><td>Consider the epsilon- NFA given by the following transition table.
<table>
<tr><th>Q\Σ</th><th>0</th><th>1</th><th>ε</th></tr>
<tr><td>→q0</td><td>q0</td><td>ϕ</td><td>q1</td></tr>
<tr><td>q1</td><td>ϕ</td><td>q1</td><td>q2</td></tr>
<tr><td>* q2</td><td>q2</td><td>ϕ</td><td>ϕ</td></tr>
</table>
Draw the transition diagram and find its equivalent NFA without epsilon moves.</td><td>2014 FALL</td></tr>
<tr><td>26.</td><td>Define regular expression. Construct a finite automata equivalent to the following regular expression.
(a + a (b+aa) * b)* a (b + aa)* a</td><td>2014 FALL</td></tr>
<tr><td>27.</td><td>State the pumping lemma for regular set. Show that $L = \{0^i 1^i \mid i>0\}$ is not regular.</td><td>2014 FALL</td></tr>
<tr><td>28.</td><td>Differentiate the Non-deterministic Finite Automata (NFA) from the Deterministic Finite Automata (DFA) with the help of the Transition Diagram (TD) and Transition Table (TT). Construct an NFA that accepts the language $L = \{a^n : n \geq 1\} \cup \{b^n a : n \geq 1\}$.</td><td>2013 SPRING</td></tr>
<tr><td>29.</td><td>Convert the NFA M=({q_0,q_1,q_2,q_3},{0,1}, δ, q_0,{ q3}) to its equivalent DFA. δ is given by
<table>
<tr><th>States</th><th>0</th><th>1</th></tr>
<tr><td>q_0</td><td>q_0, q_1</td><td>q_0</td></tr>
<tr><td>q_1</td><td>q_2</td><td>q_1</td></tr>
<tr><td>q_2</td><td>q_3</td><td>q_3</td></tr>
<tr><td>q_3</td><td>-</td><td>q_2</td></tr>
</table>
State and prove pumping lemma for regular sets.</td><td>2013 SPRING</td></tr>
<tr><td>30.</td><td>State and prove pumping lemma for regular sets.</td><td>2013 SPRING</td></tr>
<tr><td>31.</td><td>Find the regular expression for the following transition diagram

</td><td>2013 SPRING</td></tr>
</table>

<table>
<tr><td>32.</td><td>Given the transition table as follows:
<table>
<tr><th>δ / Σ</th><th>0</th><th>1</th></tr>
<tr><td>$\rightarrow q_0$</td><td>$\{q_0, q_3\}$</td><td>$\{q_0, q_1\}$</td></tr>
<tr><td>q_1</td><td>Φ</td><td>$\{q_2\}$</td></tr>
<tr><td>q_2</td><td>$\{q_2\}$</td><td>$\{q_2\}$</td></tr>
<tr><td>q_3</td><td>$\{q_4\}$</td><td>Φ</td></tr>
<tr><td>$*q_4$</td><td>$\{q_4\}$</td><td>$\{q_4\}$</td></tr>
</table>
Draw the transition diagram and also check the NFA for the input string 01001.</td><td>2013 SPRING</td></tr>
<tr><td>33.</td><td>Write Short notes on:
a) Elimination of useless symbols
b) Elimination of E-transition
c) CNF and GNF
d) Derivation tree</td><td></td></tr>
</table>

CHAPTER 3: CONTEXT-FREE GRAMMAR (CFG)

1.	For the grammar given by G=(V,∑,P,S) where, V={S} ∑={a,b,c} and P is defined as S→Xa/Yb, X→Sb/b, Y→Sa/a Design the PDA for the **following grammar.**	**2020 Fall**
2.	What is CFG? Design CFG for language L={wcw^R : w ϵ $\{a,b\}^*$ }. Test the grammar for derivation of baacaab and also draw equivalent parse tree.	**2020 Fall**
3.	What is CNF? Convert following CFG into CNF, G=(V,∑,R,S) where V={S,A,B}, ∑={a,b}, R={S→aAB|AaB|B, A→aA|ϵ, B→ab|bA}	**2020 Fall**
4.	Using the principle of Context Free Grammar, capture the expression $\left(x_1 + \frac{x_2}{x_1}\right) * (x_1 * x_2 + x_1)$and draw its parse tree.	**2019 Spring**
5.	What do you mean by Ambiguous Grammar? Explain with example. Remove the Є-production(Null) from the following grammar. $S \rightarrow ABAC$ $A \rightarrow aA/Є$ $B \rightarrow bB/Є$ C-c	**2019 Spring**
6.	Define Parse Tree. When is a grammar called ambiguous? Explain with example.	**2016 Fall, 2018 Spring**
7.	Define Chomsky Normal Form. Reduce the following grammar into CNF. *S* *bA/aB* *A*⟶ *bAA/aS/a* *B*⟶ *aBB/bS/a*	**2016 Fall,**
8.	Define language of context free grammar L (G). Write CFG for L= {w∊ {a, b}*: w has equal number of 'a' and 'b'}. Hence derive any string using same grammar and also draw parse tree.	**2019 Fall**
9.	Convert following CFG into CNF. G= (V, Σ, R, S), where V={S, A, B, a, b}, Σ={a, b}, R= {S→ASB|b, A→AbS|a|ϵ, B→SbS|A|bb}.	**2019 Fall**

10.	Describe the normal forms with suitable examples. Convert the following CFG to CNF. S ⟶ abSb/a/aAb A ⟶ bS/aAAb	2018 Spring
11.	Why ambiguities are need to be removed from ambiguous grammar? Show that following grammar is ambiguous $S \rightarrow AS \mid \in, \; A \rightarrow A1 \mid 0A1 \mid 01$	2018 Fall
12.	What is Chomsky normal form? Convert the following grammar into CNF. $S \rightarrow aAB, \; A \rightarrow aA \mid B \mid \in, \quad B \rightarrow bBc \mid \in$	2018 Fall
13.	What are applications of CFG? Write CFG for L = {w∈ {a, b}*: w is a palindrome} and also draw parse tree for the derivation of any string.	2017 Spring
14.	Define Ambiguous Grammar. Prove that following grammar is ambiguous. S→iCtS S→iCtSeS S→a C→b	2017 Spring
15.	Convert following CFG into CNF. G= (V, ∑, R, S), where V={S, A, B, a, b} ∑ = {a, b} R={S ASB|∈, A aAS|a, B SbS|A|bb}	2017 Spring
16.	Describe CNF and GNF in detail. Convert following grammar into CNF: S→bA/ aB B→bAA/aS/a C→c	2017 Fall
17.	Define ambiguity in Grammar. Check ambiguity of the Grammar: S→aB/bA A→aS/bAA/a B→aBB/bS/b	2017 Fall
18.	Define derivation tree. Differentiate between right and left derivation tree with suitable example.	2017 Fall
19.	Write a CFG for the language $L(G)=\{WW^R : W\in\{0,1\}^*\}$.	2016 Spring

20.	Consider the grammar G=(V, Σ, P, S), where, G={S,A,B} Σ={a, b} and the productions P are: S→ bA | aB A→ bAA | aS | a B→ aBB | bS | b Find an equivalent grammar in CNF.	**2016 Spring**
21.	What is ambiguous grammer? Show that given grammer is ambiguous: S →aB|ab A→aAB|a B →ABb|b	**2015 Spring**
22.	What is CNF? Convert following CFG into CNF, G=(V,∑,R,S) where V={S,A,B}, ∑={a,b}, R={S→aAB|AaB|B, A→aA|ϵ, B→ab|bA}	**2015 Spring**
23.	What is CFG? Design CFG for language $L(G)=\{a^m b^n : m>=n\}$ along with parse tree.	**2015 Fall**
24.	State CNF and GNF. Convert following CFG into CNF, G=(V,∑,R,S) where V={S,A,B,C,a,b,c}, ∑={a,b,c}, R={S→ABA|abA|BC, A→aA|ϵ, B→baB|c, C→aC}	**2015 Fall**
25.	Define Parse Tree. When is a grammar called ambiguous? Explain with example.	**2014 Spring**
26.	Describe the normal forms with suitable examples.	**2014 Spring**
27.	Convert the following CFG to CNF. $S \longrightarrow abSb/a/aAb$ $A \longrightarrow bS/aAAb$	**2014 Spring**
28.	Explain Chomsky classification of language. If G is the grammar S→ SbS | a, show that G is ambiguous.	**2014 Fall**
29.	Show that the grammar S→aB| ab, A→aAB | a, B→AB | b is ambiguous.	**2014 Fall**

30.	Reduce the following CFG to Chomsky Normal Form. $S \quad aB/bX$ $A \longrightarrow Bad/bSX/a$ $B \longrightarrow aSB/bBX$ $X \longrightarrow SB/aBx/ad/B$	**2013 Spring**
31.	When a grammar is called ambiguous? Prove that the following grammar is ambiguous. $S \longrightarrow AB/aaB$ $A \longrightarrow a/Aa$ $B \longrightarrow b$	**2013 Spring**
32.	How do you identify the word w over alphabet $\sum$ is generated by a given CFG? Illustrate.	**2013 Spring**
33.	When the grammar is ambiguous? Show that the given grammar is ambiguous: S->aB/ab, A->aAB/a, B->ABb/b.	**2013 Fall**
34.	How do you define the sentence and the sentential form in a CFG? Convert the grammar with the following set of production rules into Chomsky Normal Form (CNF). $S \rightarrow ABaC$ $A \rightarrow BC$ $B \rightarrow b\backslash\lambda$ $C \rightarrow D\backslash\lambda$ $D \rightarrow d$	**2013 Fall**
35.	Reduce the following grammar G to CNF. $S \rightarrow aAD$, $A \rightarrow aB/bAB$, $B \rightarrow b$, $D \rightarrow d$	**2013 Fall**

CHAPTER 4: PUSHDOWN AUTOMATA (PDA)

1.	In what aspect PDA is stronger than finite automata? State closure properties of context free grammar.	**2020 Fall**
2.	State the Pumping lemma for context free language. Prove that the language $L=\{0^n1^n2^n \mid n>=0\}$ is not context-free language.	**2020 Fall**
3.	What is instantaneous description of PDA? Design a PDA which accepts the language $L=\{w\epsilon\{0,1\}^* :$ w has equal number of 0's and 1's$\}$.	**2019 Spring**
4.	Write about closure properties of context free language.	**2019 Spring**
5.	State the Pumping lemma for context free language. Prove that the language $L=\{a^nb^nc^n \mid n>=0\}$ is not context-free language.	**2019 Spring**
6.	Write a brief description of push down automata, with suitable example.	**2016 Fall**
7.	Design a Pushdown Automata which accepts the language L = {w Є {a, b}* / w has equal number of a's and b's}.	**2016 Fall**
8.	State pumping lemma for CFL. Show that the language $L=\{ a^nb^nc^n /$ $n>=0\}$is not a context free language.	**2016 Fall**
9.	"PDA is stronger than FA". Explain this statement.	
10.	In what aspect PDA is stronger than finite automata? State closure properties of context free grammar	**2019 Fall**
11.	State the Pumping lemma for context free language. Prove that the language $L= \{0^n1^n2^n / n>=0\}$ is not context-free language.	**"**
12.	How to find a PDA from a CFG? Construct a PDA from the grammar: $S\to 0S1S$, $S\to 1S0S$, $S\to \varepsilon$	**2018 Spring**
13.	Give the formal definition of pushdown automata. Construct a PDA accepting the language $L = \{0^n1^n \mid n > 0\}$	**2018 Fall**
14.	State pumping lemma for context free language. Prove that language $L = \{a^nb^nc^n \mid n > 0\}$ is not context free language.	**"**
15.	Define push down automata. Explain equivalence of push down automata with context free grammar with suitable example.	**2017 Spring**
16.	Design a PDA that accepts those strings "having total number of a equal to the sum of number of b and c with sequence of a,b,c (i.e. $a^ib^jc^k$: i=j+k). Hence test your design for the string "aaaabbcc".	**"**
17.	Explain properties of context free language. Show context free languages are closed under union.	**"**
18.	Define push down automata. Design a PDA for the following language, $L = \{a^nb^{2n} : n > 0\}$.	**2017 Fall**

19.	State the Pumping lemma for context free language. Explain about decision properties of CFLs.	“
20.	Why PDA is stronger than FA? State the closure properties of context free languages.	“
21.	Define push down automata. Design a PDA for the following language, $L = \{a^n b^{2n} : n > 0\}$.	2016 spring
22.	Design a Pushdown automata which accepts the language $L=\{w \in \{a,b\}^* / w$ has equal number of a's and b's$\}$.	2015 Spring
23.	Sate pumping lemma for CFL's. Mention the closure properties of CFL's.	“
24.	Formally define a PDA. Design a PDA which accepts the language $L=\{w \in \{x,y\}^* :$ w has equal number of x's and y's$\}$.	2015 Fall
25.	State pumping lemma for context free language. Show that $L= \{ a^n b^n c^n \mid n>=1 \}$ is not context free language.	“
26.	Prove that the family of context free language is not closed under intersection and complementation.	2014 Spring
27.	Explain the decision algorithms for context free languages.	“
28.	Why was Pushdown Automata (PDA) introduced? Construct a PDA that accepts the language $L=\{x^i y^j z^k : i,j,k>=0$ and $i=k$ or $j=k\}$.	2014 Fall
29.	State and prove pumping lemma for Context Free Language (CFL).	“
30.	Check if the language $L=\{0^m 1^m : m>=0\}$ is regular or not.	“
31.	Construct a Push Down Automata for the following language. $L = \{a^n b^{n+1} / n = 1, 2, 3.........\}$	2013 Spring
32.	Only state the pumping lemma for CFL. Show that the language $L=\{ a^n b^n c^n / n>=1\}$ is not a context free language.	“
33.	Show that any the Context Free Language (CFL) is closed under union, concatenation and star-closure.	2013 Fall
34.	Show that the language $L=a^n b^n c^n / n \geq 1$ is not context free.	“
35.	Write Short Notes on: a) Pumping Lemma for CFL b) PDA c) Closure properties of CFL d) Instantaneous description of PDA	

CHAPTER 5: TURING MACHINE

1.	Design a Turing machine that transforms #w# to ##w#. Where # represents blank symbol and w is any string of a and b.	2020 Fall
2.	What is configuration of Turing machine? Show that f(x) = x+1 is Turing computable.	"
3.	How can you represent a Turing Machine? Show that the function, f (n) = x+1, is Turing computable.	2019 Spring
4.	Design a Turing Machine as a right shift machine which transforms #w# into ##w# with alphabet $\sum=\{a,b,\#\}$.	"
5.	What is a Turing Machine? Describe any three types of Turing Machine.	2016 Spring
6.	Define Turing machine. Design a Turing machine that accepts the language $L=1^n2^n3^n \mid n>=0$.	2019 Fall
7.	How can you represent turing machine for computing a function? Show that the function f(n)=n+1, is turing computable.	2019 Fall
8.	Define Turing machine. Design a Turing machine that accepts the language $L=1^n2^n3^n \mid n>=0$.	2018 spring
9.	How can you represent turing machine for computing a function? Show that the function f(n)=n+1, is turing computable.	2018 spring
10.	How can you represent a Turing Machine? Show that the function, f (n) = x+1, is Turing computable.	2018 Fall
11.	Describe about Universal Turing machine. What is the application of Turing Machine?	"
12.	How can you represent a Turing Machine? Show that the function, f (n) = n+1, is Turing computable.	2017 Spring
13.	Design a Turing machine that works as a simple eraser, which changes every non-blank symbols to blank with alphabet $\sum$ = {0, 1, #}. Hence test your design for #0101# to ######.	"
14.	How can you represent a Turing Machine? Describe about Universal Turung Machine.	2017 Fall
15.	Design a Turing machine for computing a function f(w)=w#w.	"
16.	Design a Turing machine which compute the function f(m) = m+1 for each m that belongs to the set of natural numbers.	2016 Spring
17.	How can you represent Turing machine for computing a function? Show that the function f(n)=n+1, is Turing computable.	2015 S

18.	Define Turing machine. Design a Turing machine that accepts the language $L=1^n2^n3^n \mid n>=0$. ... enumerable language? Show that	"
19.	Define Turing machine. Design a Turing machine that accepts the language $L=1^n2^n3^n \mid n>=0$.	**2015 F**
20.	What is K-tapes Turing machine? Show that any K tapes Turing machine can be converted to an equivalent one tape Turing machine.	"
21.	What is a Turing Machine? Describe any three types of Turing Machine.	**2014 S**
22.	Construct a Turing Machine that recognizes the language L= { $p^nq^mr^n$:m, n>=0}.	**2014 F**
23.	Describe the extensions of Turing Machine.	"
24.	Dsitinguish TM from the FA and the PDA.	**2013 S**
25.	Design a Turing machine that accepts the language. $L = 1^n2^n3^n / n \geq 0$	"
26.	Write short notes on: a) K-tape turing machine b) Representation of TM by Instantaneous description	

CHAPTER 6: UNDECIDABILITY

1.	Write about church turing thesis and universal turing machine.	**2020 Fall, 2019 Spring**
2.	What is recursive and recursive enumerable language? Write some properties of recursively enumerable language.	**2016 Spring**
3.	Describe Church's Hypothesis. Also, illustrate your understnading of Halting problem.	
4.	What is recursive and recursively enumerable language? Show that the union of two recursive language is also recursive.	**2019 F, 2015 S**
5.	Write about church turing thesis and universal turing machine.	**2019 F, 2018 F, 2015 S**
6.	Differentiate between Recursive & Recursively enumerable language. Explain the recursive properties of language in detail.	**2018 Fall**
7.	Define universal Turing machine and explain its encoding technique in detail with suitable example.	**2017 Spring**
8.	Explain the properties of recursive and recursive enemurable languages.	**2017 Fall, 2016 Spring**
9.	Explain space complexity and time complexity with suitable examples.	**2016 spring**
10.	What is recursive and recursively enumerable language? Show that the complement of recursive language is also recursive.	**2015 F**
11.	Write about Church-Turing thesis. Explain about encoding of turing machine.	"
12.	Discuss the Recursive function theory. Prove that the union of two recursive languages is recursive.	**2014 S**
13.	What is undecidibality in computation? Explain about undecidable problems of TM.	**2014 F**
14.	Differentiate between Recursive and Recursively enumerable languages.	
15.	Explain briefly about recursive and recursively enumerable language.	**2013 S**
16.	What are two computational complexities that matter in designing efficient program?	**2013 F**
17.	State and Illustrate the Church Turing Hypothesis.	"
18.	Differentiate the Recursive Language from the Recursively Enumerable Language.	"
19.	Write short notes on: a) Universal TM b) The Halting problem c) Church's Thesis	

CHAPTER 7: COMPLEXITY THEORY

1.	Explain in brief the P and NP complete problems with suitable examples.	**2020 F, 2015 S**
2.	How does computability differ from complexity theory? Describe about the time and space complexity.	**2019 Spring**
3.	Compare and contrast class P and class NP problems with examples for each.	**016 Spring**
4.	State computational complexity theory? Explain class NP with suitable example.	**2019 Fall**
5.	Define computability theory. Differentiate between P complete problem and NP complete problem with example. Does P problem equals to NP?	**2018 Spring**
6.	How does computability differ from complexity theory? Describe about the time and space complexity.	**2018 Fall**
7.	What are P, NP and NP-Complete problems? Explain with examples.	"
8.	Explain in brief the P and NP complete problems with suitable examples.	**2017 S, 2015 S**
9.	Differentiate between tractable and intractable problems. Also write some examples of NP completeness problems.	**2017 F, 2016 S**
10.	Write about computational complexity theory. What are tractable and intractable problems?	**2015 F**
11.	What are P, NP and NP-Complete problems? Explain with examples.	"
12.	Describe the Computational Complexity theory.	**2014 S**
13.	What are tractable and intractable problems? Explain the NP complete problems with suitable examples.	"
14.	Define class P and class NP. What is NP-completeness?	**2014 F**
15.	Explain in brief the P and NP complete problems with suitable examples.	**2013 S**
16.	What are two computational complexities that matter in designing effieient program? Explain them.	"
17.	What are the NP-Hard and NP-Complete problem? Illustrate.	**2013 F**
18.	Write short notes on: a) Undecidability b) Computable languages c) Big O Notation d) Time and Space complexity	