

Subject: _____ No: _____
Date: _____/_____/_____
Defining Languages
• A language \rightarrow A set of strings of symbols
• A grammar \rightarrow The rules for forming the strings in a language
• If a C++ program is one long string of characters, the language of C++ programs is defined as
$$C++ \text{ Programs} = \{ \text{strings w.w.t.s a syntactically correct C++ program} \}$$

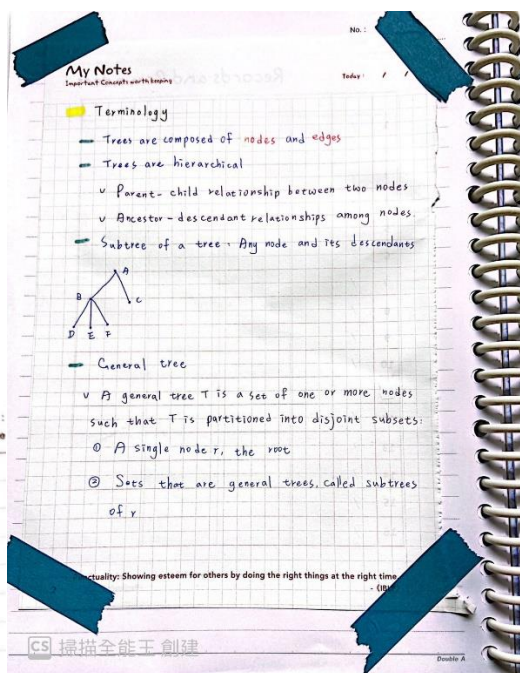
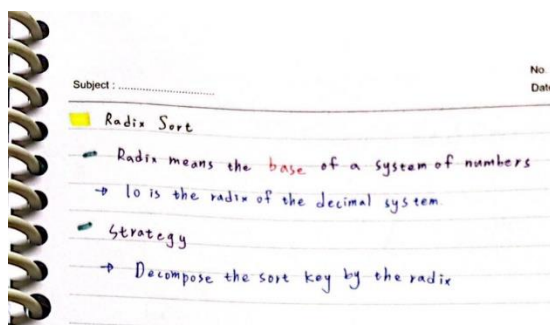
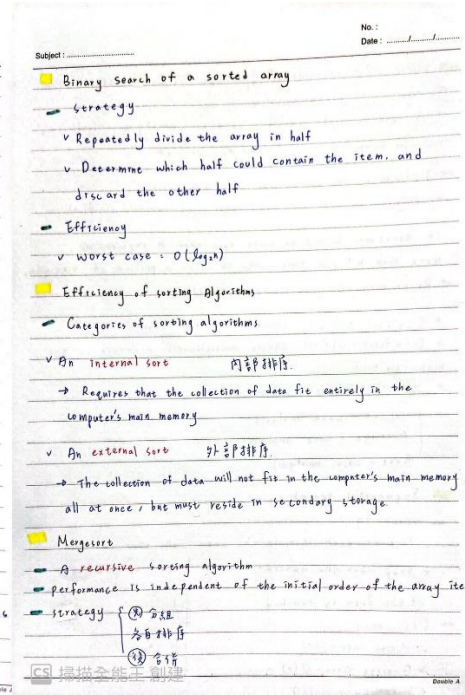
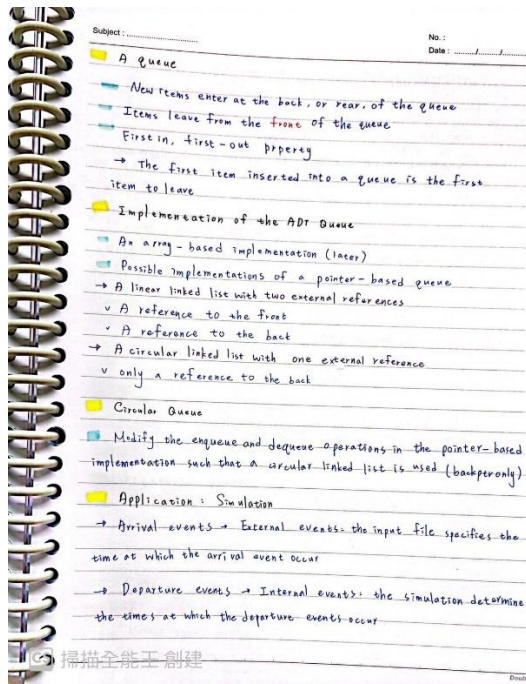
The Basic of Grammars
• A recognition algorithm sees whether a given string is in the language
 \hookrightarrow A recognition algorithm for a language is written more easily if the grammar is recursive

Algebraic Expressions
• Infix expressions \rightarrow An operator appears between its operands
Example: $a+b$
• Prefix expressions \rightarrow An operator appears before its operands
Example: $+ab$
• Postfix expressions \rightarrow An operator appears after its operands
Example: $ab+$

Subject: _____ No: _____
Date: _____/_____/_____
Algebraic Expressions
• To convert a fully parenthesized infix expression to a prefix form
 \hookrightarrow Move each operator to the position marked by its corresponding open parenthesis
 \hookrightarrow Remove the parentheses
Example { Infix expression: $(a+b)*c$
Prefix expression: $*+abc$
• To convert a fully parenthesized infix expression to a postfix form
 \hookrightarrow Move each operator to the position marked by its corresponding closing parenthesis
 \hookrightarrow Remove the parentheses
Example { Infix expression: $(a+b)*c$
Prefix expression: $ab+c*$
• Advantages of prefix and postfix expressions
 \hookrightarrow No precedence rules
 \hookrightarrow No association rules
 \hookrightarrow No parentheses
 \hookrightarrow Simple grammars algorithms
 \hookrightarrow Straightforward recognition and evaluation

Subject: _____ No: _____
Date: _____/_____/_____
Recognizing String in a Language
 $\hookrightarrow L = \{ w \&w' \mid w \text{ is a possibly empty string of characters other than } \$, w' = \text{reverse}(w) \}$ 逆序
 \hookrightarrow A solution using a stack
 \rightarrow Traverse the first half of the string, pushing each character onto a stack. push: 压入
 \rightarrow Once you reach the \$, for each character in the second half of the string, match a popped character of the stack. pop: 弹出
Evaluating Postfix expression
 \hookrightarrow A postfix calculator
 \rightarrow When an operand is entered, the calculator (push it onto a stack)
 \rightarrow When an operator is entered, the calculator (Applies it to the top two operands of the stack)
(pops the operands from the stack)
(pushes the result of the operation onto the stack)
Converting Infix Expressions to Equivalent Postfix Expression
 \rightarrow Append an operand to the end of an initially empty string

Subject: _____ No: _____
Date: _____/_____/_____
Non-recursive Solution That Use a Stack
 \hookrightarrow The solution performs an exhaustive search
 \rightarrow Beginning at the origin city, the solution will try every possible sequence of flights until either
(It find a sequence that goes to the destination city)
The Relationship Between Stacks and Recursion
 \rightarrow Typically, stacks are used by compilers to implement recursive methods.
(During execution, each recursive call generates an activation record that is pushed onto a stack)
 \rightarrow Stacks can be used to implement a non-recursive version of a recursive algorithm



My Questions

Problems & Difficulties needing exploration

- Parent of node B
 - The node directly above node B in the tree
- Child of node B
 - A node directly below node B in the tree
- Root
 - The only node in the tree with no parent
- Subtree of node B
 - A tree that consists of a child (if any) of node B and the child's descendants

My Opinions

Thoughts, inspirations, and suggestions

- Leaf
 - A node with no children
- Siblings
 - Nodes with a common parent



CS 掃描全能王 創建

守時：在對的時間，做對的事。來表明對別人的尊重。

- (後基)

3

My Notes

Important Concepts worth keeping

Today: / /

- Ancestor of node B
 - A node on the path from the root to B
- Descendant of node B
 - A node on a path from B to a leaf

Binary Tree

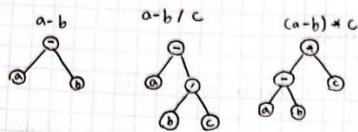
- A binary tree is a set T of nodes such that either
 - ✓ T is empty, or
 - ✓ T is partitioned into three disjoint subsets:
 - ① A single node r , the root
 - ② Two possibly empty sets that are binary trees, called the left subtree of r and the right subtree of r .

We are convinced that love is the principal guiding force in education. We, teachers and students alike, pursue mutual growth through instruction by both words and deeds, in a spirit of love and respect for one another.

- (CYCU Education Philosophy)

My Questions

Problems & Difficulties needing exploration



- Height of a tree
 - Number of nodes along the longest path from the root to a leaf

My Notes

Important Concepts worth keeping

Today: / /

Full Binary Trees

- A binary tree of height h is full if
 - Nodes at levels $< h$ have two children each
- Recursive definition
 - If T is empty, T is a full binary tree of height 0
 - If T is not empty and has height $h > 0$, T is a full binary tree if its root's subtrees are both full binary trees of height $h-1$

Complete Binary Trees

- A binary tree of height h is complete if
 - It is full to level $h-1$ and
 - Level h is filled from left to right

Faith, Hope and Love!
Faith lays the foundation,
Hope brings sunshine,
Love changes everything.
- Sunday



My Questions

Problems & Difficulties needing exploration

- A binary tree of height h is complete if
 - All nodes at levels $< h-2$ have two children each and
 - When a node at level $h-1$ has children, all nodes to its left at the same level have two children each, and
 - When a node at level $h-1$ has one child, it is a left child

Balanced Binary Trees

My Opinions

Thoughts, inspirations, and suggestions

- A binary tree is balanced if the heights of any node's two subtrees differ by no more than 1
- Complete binary trees are balanced
- Full binary trees are complete and balanced



陽光使人健康
信實可靠打地基
樂觀希望收發正能量
真心願望可以改變一切
- 三第 7

CS 掃描全能王 創建

My Notes

Important Concepts worth keeping

Today: / /

Representations of Binary Tree

- An array-based representation
 - Uses an array of tree nodes
 - Requires the creation of a free list that keeps track of available nodes
- A pointer-based representation
 - Nodes have two pointers that link the nodes in the tree

Traversals of a Binary Tree

- Preorder traversal
 - Visit root before visiting its subtrees (Before the recursive calls)
- Inorder traversal
 - Visit root between visiting its subtrees (Between the recursive calls)

ONCE, all the villagers decided to pray for rain. On the day of prayer all the people gathered, but only one boy came with an umbrella.
That's FATE!

- Anonymous

Subject

My Questions

Problems & Difficulties needing exploration

- Post order traversal
 - Visit root after visiting its subtrees (After the recursive calls)

No.:

Date: