unit-1

Q. Define Data structures explain the importance of data structure in computer science.

⇒ A data structure is a special format for organizing, processing, retrieving and storing data. There are several basic and advance type of data structure

Q. importance of Data structure

Data structure are essential for main reasons, they make the code more — efficient and then make the code easier to understand

Q Explain difference between primitive & non-primitive data structure provide Example

primitive	Non primitive
i) They are kind of data structure that store data of only one type	i) They one kind of data in that can store more than one type one
ii) The size depends on the type of data	eg. case of non primiti data structure
iii) they can be used to calls method	iii) they cannot the used to call methode.

Q. Explain the concept of an array

Sol Declaration

```
int arr[] = {1,2,3,7,5};
```

initialization with zero

```
int arr[] = {0};
```

initializing an array

```
int Demo_array[5];
int i;
for(i=0; i<5; i++)
Demo array[i] = { }
}
```

Q- Describe the concept of linked list and Array

Sol They are many of linked list

i) singly linked list
ii) doubly linked list
iii) circular linked list
iv) circular doubly
v) header linked list

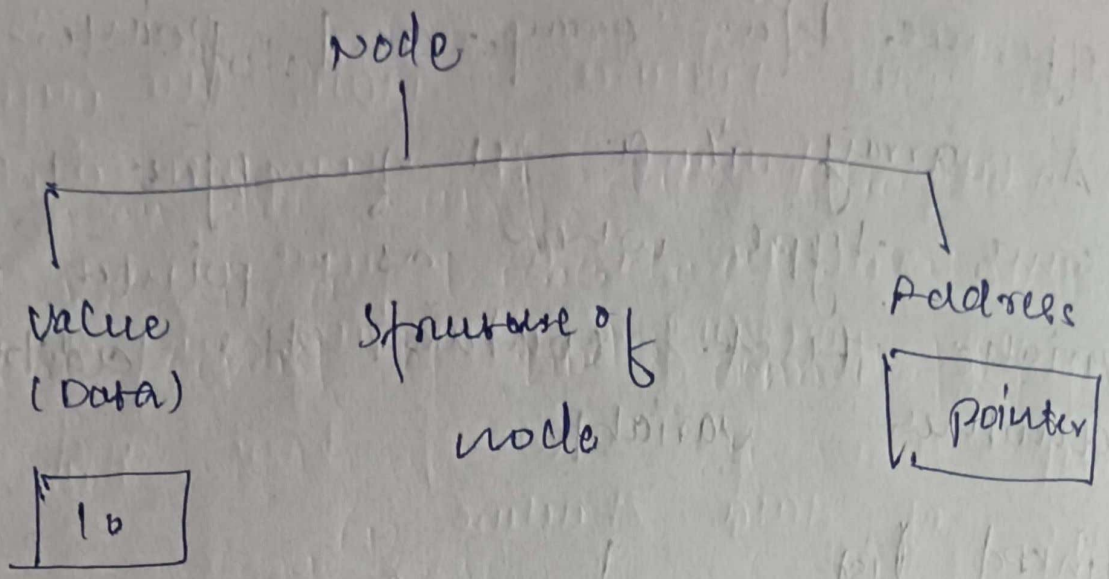
Q. Difference b/w array and pointer

Sol. An array is a set of object of the same type where as a pointer is a variable that contains the address of another variable

Linked list	Array
Data structures	Data structure
non-contiguous	contiguous
Memory allocation	Memory allocation
Typically allocated to one by one	Typically allocated
Access sequence	Access same order

Q. Explain structure of node

Sol. A linked list is a way to store a collection of element with an array. These can be character or integer. Each element is linked list is stored in form



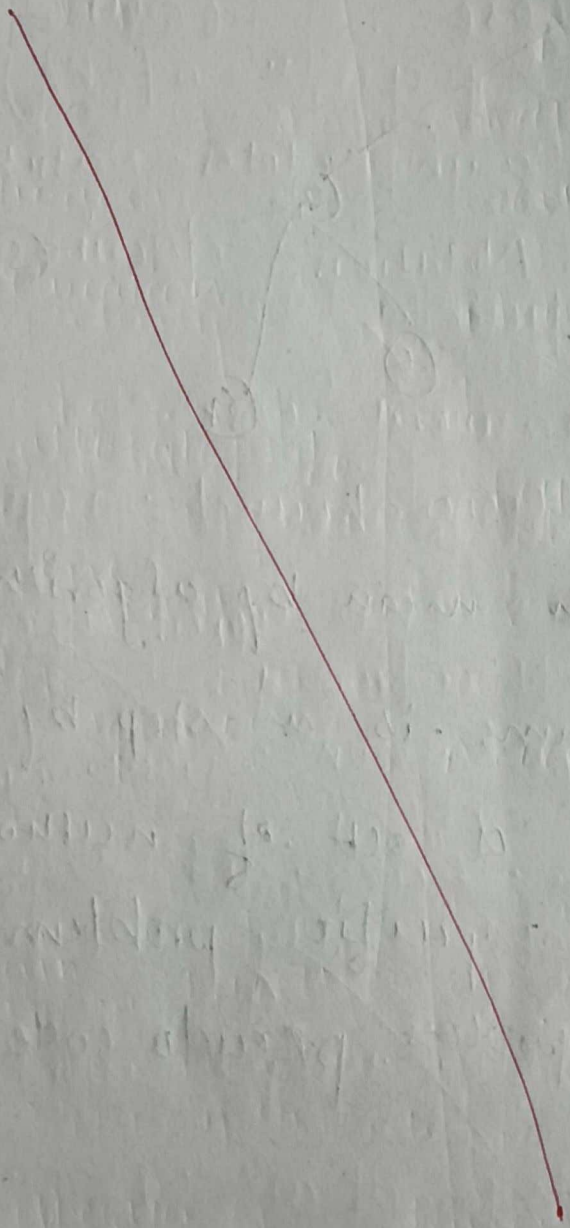

```
struct linked list {
    int data
    node * next;
}
void add list (Node * N, head, int value)
    node * temp;
    new Node → data = value;
    new node next = Num -
while (P == ln rev)
    return true;
```

Q. Describe two different type of linked list

1) singly: it is a unidirection linked list

ii) Doubly:- It is a bidirectional linked list

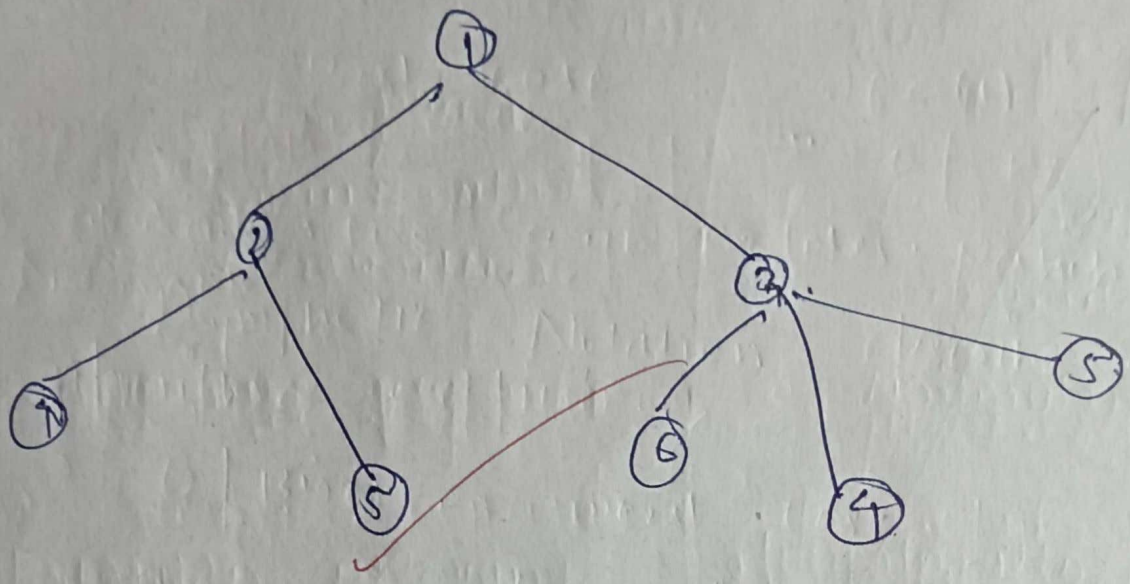
iii) Circular:- It is a circular linked list



Tutorial - 2

① Explain tree and data structure

A tree is a hierarchical data structure that consist of nodes connected by edges



Q. What are you mean by algorithms

Ans An algorithms is a step by step process. or a set of methode design to refer a specific problem or data such as ~~process~~ pseudo code or flow chart

Q. Distinguish between Algorithm and pseudo code

Sol Algorithms:-

a formalities set of instruction to solve a problem to writing algorithms human understanding

Q) Define space and time complexity

Space complexity:- It refers to the amount of space and memory required

Time complexity:- It refers to the computation time required to an algorithms

Q.) Explain the worst count. but and avg space of algorithms. array with example

Sol

. Worst case:- The maximum time & space of an algorithms taken even under the most challenging input.

Best case:- The minimum ~~cost~~ time or space of an algorithms taken typically under the most features. input

Avg case: The extra time by a span complexity of an algorithms over all possible input

Eg: in binary search:

worst case — O(n)

Best case — O(1)

Avg case — O(n/2)

Q. Discuss the asymptotes notation

Ans Asymptotic Notation describe an algorithms efficiency

→ O (Big O) — worst case upper bound required

→ O (theta) → no upper case and bound.

⇒ O (Big O) :- non-tight upper bound for

n^2 — not exactly

ω (little omega) - non-tight

Tutorial - 3

Q. what is an abstract data type? what are still not converted in an ADT

An adt is a data type define by behaviour (semantics) rather than is implementation its input. what operate can be performed on the data and what the ~~expected~~ behaviour of the logican should be without specifying how these operator can implements

Q. list out area in which data structure are applied. Extensively

1. An operating system (eg. many management process, schedulers)
2. Database (eg. indexing, query, process)
3. Difficult implementation (eg graphs for row. network, discision. Item.
4. vector data structures

Q. write down ~~these~~ step. to. modify a node in linked list.

(i) Traverse the linked list until you reach the node to be modified

(ii) Access the data part of it

(iii) Proceed to update the data

iv) Easier the list (if any) to adjust node remain unchange

Que: state the properties of list abstract data type with suitable example

Dynamic size data type

(1) Limited collection: the element in list have same specific only

(2) Duplicate allocation: A list can contain duplicate element

(3) Sequential Access in sequence

Q what is the advantage of an ADT?

⇒ The advantage of an ADT is that it allows of the what operation can be performed this enable code modularity, reusability and can of maintance since the underlying implementa

Q Explain the insertion operation in linked list from it's node inserted after a specified node

Sol

① Create a new node

② set the node. next. point to the node behind it

③ update the specific node next pointer to the new node

Q. Define the deletion operation from a linked list

i) traverse the list to find the new node just before the one to be deleted

ii) change the next ptr of the previous node to be pointed after the delete

iii) free the memory allocated to the node to be deletion

Tutorial - 4 3/10/24

Q. What is the length of the path in a tree?

The length of a path in a tree is the number of edges between two nodes for a root to leaf path, it is the number of edges from the root node to leaf node

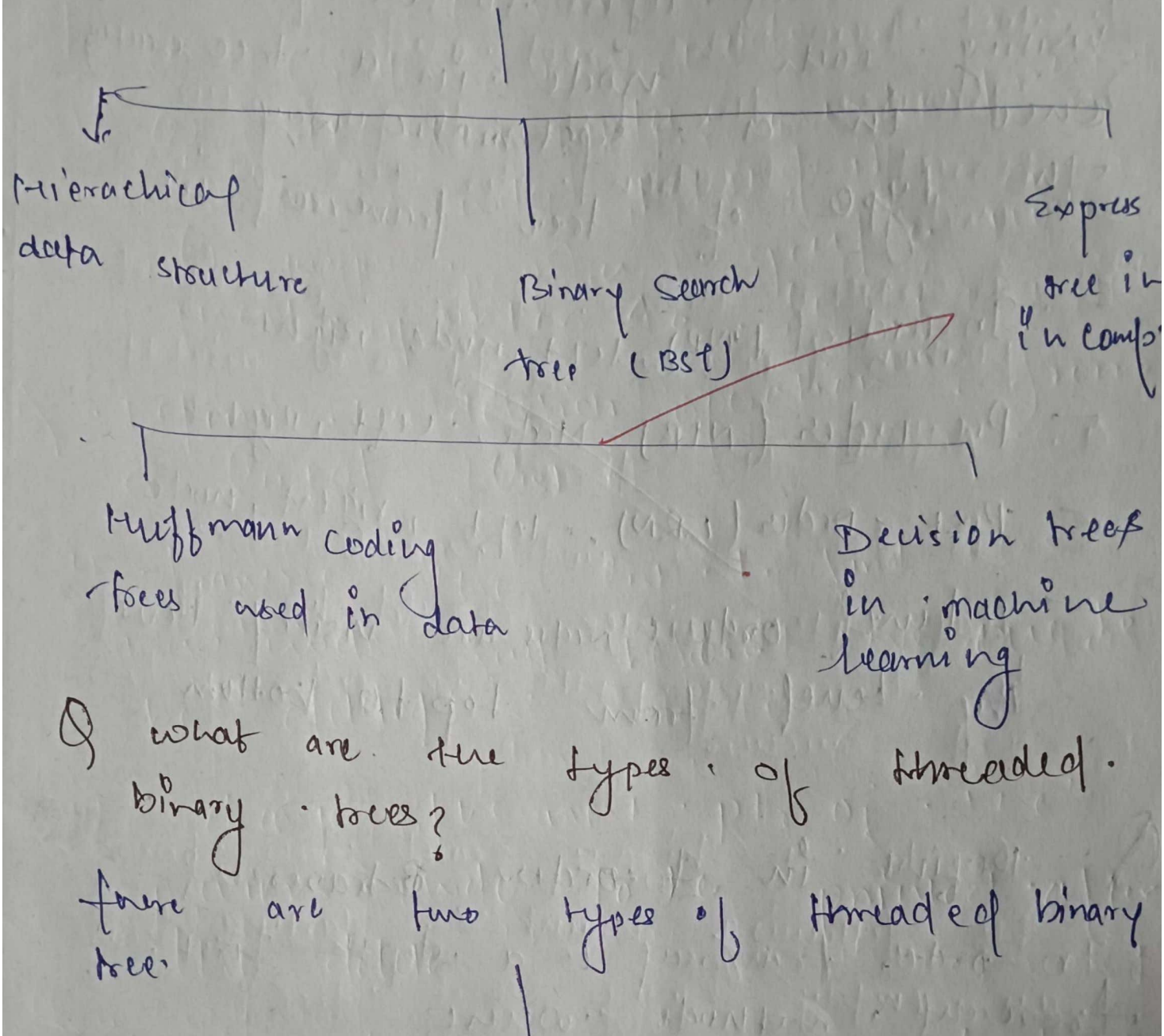
Q. What are the two methodes of Binary tree implementation

The two common methodes of implementing a binary tree are.

- Linked Representation
  - ⇒ each node contains data and pointers to its left and right children
- Array Representation
  - ⇒ the tree is stored as an array where the root is at index 0 the left child of node i is at 2i+1 & right is 2i+2

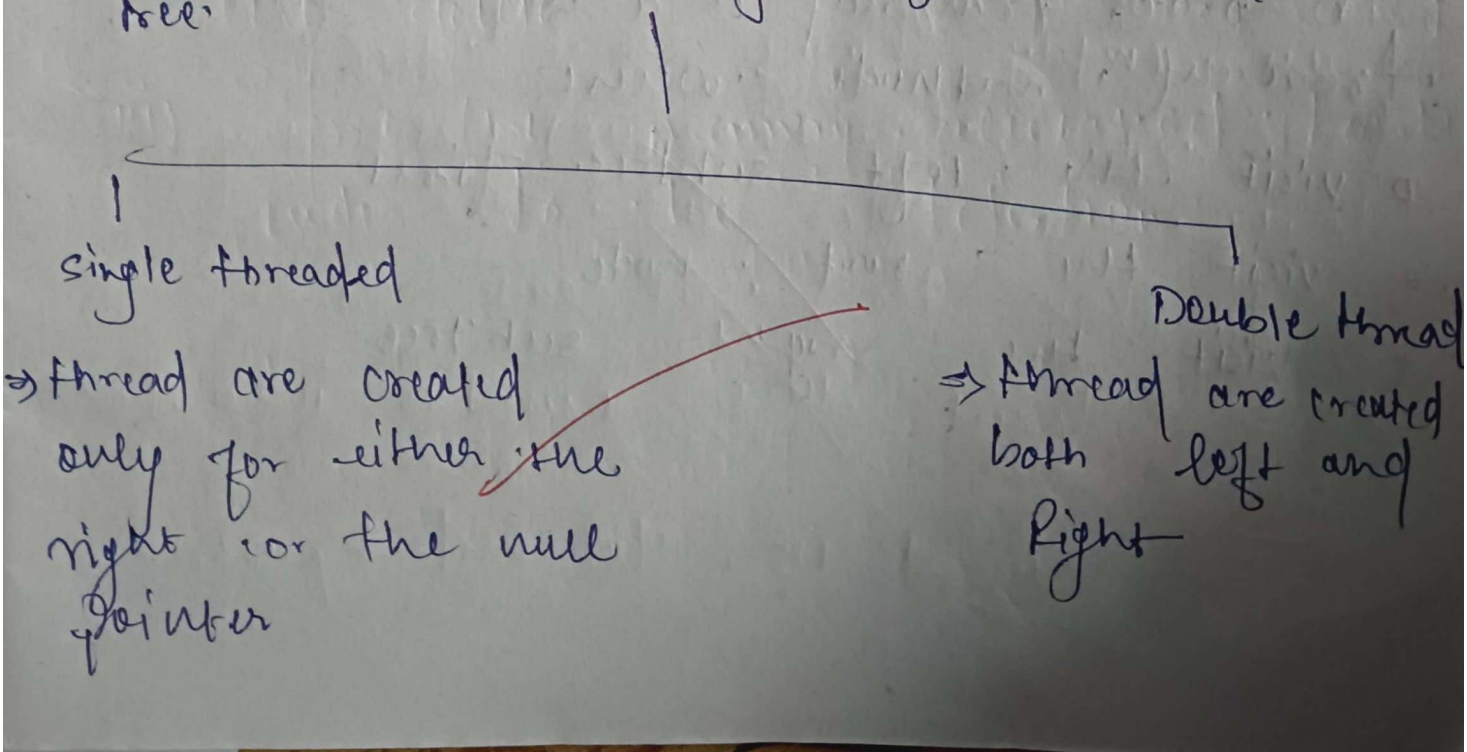
Q. what are the application of Binary tree?

Binary trees are used in various application such as.



Q what are the types of threaded binary trees?

There are two types of threaded binary tree.



Q Define tree traversal and mention the types of traversing

⇒ tree traversal refers to the process of visiting each node in a tree exactly once in a systematic way

the type of tree traversal are:

- In-order (LNR), left, node, right
- Pre-order (NLR), Node, left, right
- Post-order (LRN), left, right, node
- Level order: Nodes are visited level by level from top to bottom

Q. Define in In-order traversal.

In-order traversal is a depth-first traversal method where:

- visit the left subtree first
- visit the root node
- visit the right subtree

Oct 19, 2024, 12:04

Q:- Explain the AVL tree - insertion and deletion with a suitable example

- An AVL tree is a self-balancing binary search tree where the heights of two child subtrees of any node differ by at most one
  - Insertion
  - deletion

e.g:- After inserting element 20, 4, 26, 3, 9 into an AVL tree, rotation may be needed to keep the tree balanced

Que:- Explain the B-trees with a suitable example

- A B-Tree is a self-balancing tree data structure that maintains sorted data and allows searches, sequential access, insertions and deletions in logarithmic time. It is a generalisation of an binary tree in that a node can have more than two children

e.g B-Tree are used in databases and file system where multiple key can be stored in one node and nodes are adjusted to maintain balance upon insertion or deletion.

Q. write an algorithm for inserting a node in a threaded Binary tree

Algorithm:-

1. start from the root node.
2. Traverse the tree to find the correct location for the new node
3. Insert the node at the identified position (either left are right)
4. Adjust the threads
5. if the new node is inserted as a leaf; its pointer should be threaded to the appropriate in-order predecessor and successor

Q. Create a binary search tree for the following numbers starting from an empty binary search tree (45, 26, 10, 60, 70, 30, 40) Delete key 10, 60 one after one the another and show tree at each stage

Step-1: Insert the number into the BST:

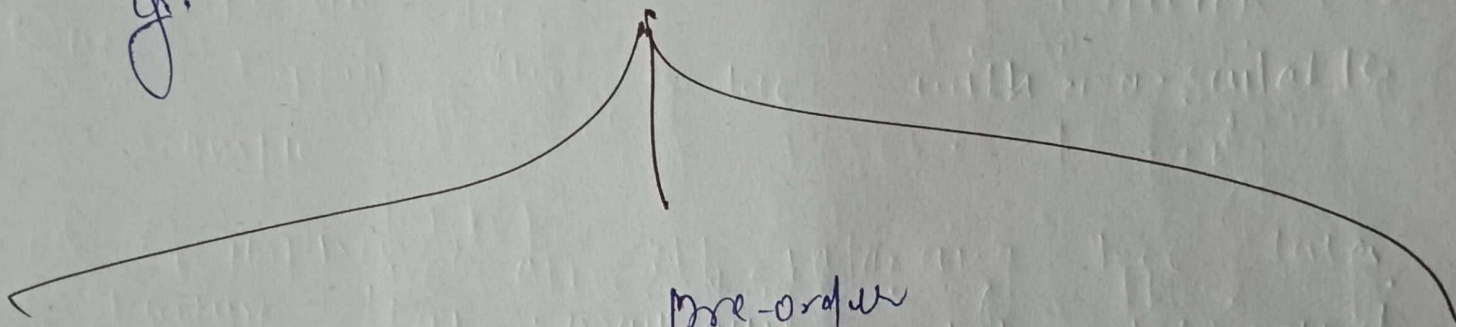
=> First insert 45, followed by 26, 10 60, 70, 30, 40

Step-2 :- Delete 10, then 60, after each deletion, adjust the tree to maintain the Binary search property

Q. Construct an expression tree for the expression (a+b) * (c-d+e) * f) * g

Give the output:

Exp. tree: The tree will have star (*) as the root node and main subtrees for (a+b), (c-d+e), and f, followed by g.



In-order traversal

a+b*c-d+e*f*g

pre-order traversal

***+ab+-cdefg

post Order traversal

ab+cd-e+f*g**

Finally 15 to 24

A+