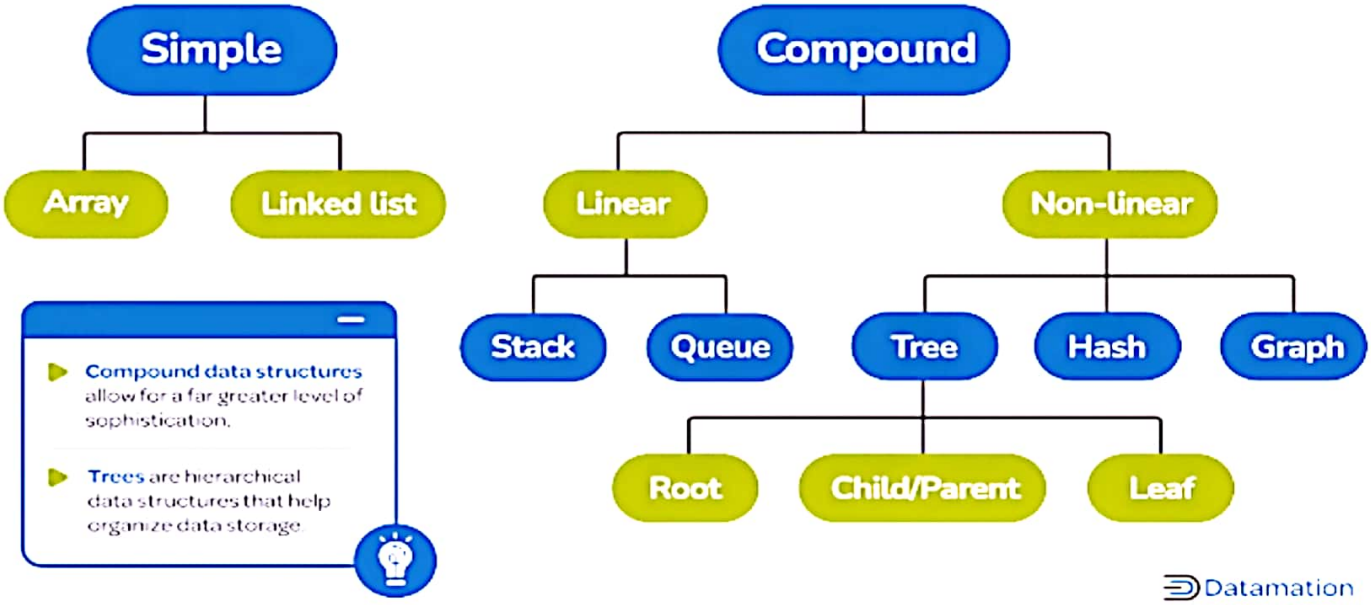
Data Structure & Algorithm

Types of Data Structures



ABIR

MAPI Computer Technology (2022 - 2023)

Sorry for spelling mistakes and truncation

CamScanner

অতিসংক্ষিপ্ত :-

১। Data বলতে কী বুঝায় ?

সাধারন অর্থে Data বলতে কোন কিছুর মানকে বুঝায়।

২। ইনফরমেশন বলতে কী বুঝায় ?

Data কে প্রক্রিয়া করনের পর যে অর্থপূর্ন মান পাওয়া যায় তাকে Information বলে।

৩। Data Structure বলতে কী বুঝায় ?

একটি নির্দিষ্ট Data সংগঠনের Logical বা Mathematical মডেলকে Datastructure বলে।

৪। Data structure এর কিছু অপারেশনের নাম লেখ।

(i) Traversing. (ii) Searching. (iii) Sorting.
(iv) Inserting. (v) Deleting. (vi) Marging.

৫। অ্যালগরিদম বলতে কী বুঝায় ?

সমস্যা সমাধানের নির্দিষ্ট ধাপ সমূহকে ক্রমানুসারে সাজানোকে অ্যালগরিদম বলে।

৬। pseudo code বলতে কী বুঝায় ?

অ্যালগরিদমে লিখিত সংক্ষিপ্ত ও সহজবোধ্য নির্দেশমালা বা statement সমূহকে programable Design language PDL বা pseudo code বলে।

৭। অ্যালগরিদম নোটেশন বলতে কী বুঝায়?

Algorithm এর জন্য বিধিবদ্ধ কিছু সাংকেতিক চিহ্নকে অ্যালগরিদম নোটেশন বলে।

৮। Structured programming বলতে কী বুঝায়?

বিভিন্ন হাই-লেভেল ল্যাংগুয়েজ (C, C++, pascal, Java) ইত্যাদি ভাষায় লিখিত program কে Structured programming বলে।

৯। লিনিয়ার Data structure বলতে কী বুঝায়?

এক জাতীয় Data সমূহ যদি পর্যায়ক্রমে সাজানো থাকে তখন তাকে লিনিয়ার Data structure বলে।

১০। Traversing বলতে কী বুঝায়?

Computer memory তে সংরক্ষিত যেকোন Data পরিদর্শন করাকে Traversing বলে।

১১। pointer Array কাকে বলে?

pointer সমূহকে যখন একটি Array এর মাধ্যমে সজ্জিত করা হয় তখন তাকে pointer Array বলে।

১২। Jagged array কাকে বলে?

যে সমস্ত Array এর Row অথবা Column Data প্রদান দিয়ে শুরু হয় কিন্তু অব্যবহৃত স্টোরেজ লোকেশন দিয়ে শেষ হয় তাকে Jagged array বলে।

১৩। এক জাতীয় Data উপাদানের Data সমূহকে Data ফিল্ড বলে।

১৪। Record বলতে কী বুঝায়।

ফিল্ড এর সংরক্ষিত Data কে রেকর্ড বলে।

১৫। STACK কী?

STACK হলো এমন এক ধরনের Data structure যার কেবলমাত্র END থেকে Data সংযোজন বা বিয়োজন হয়ে থাকে।

১৬। PUSH এবং POP বলতে কী বুঝায়?

PUSH : STACK এ কোন আইটেম সংযোজন করাকে PUSH বলা হয়।

POP :- STACK এ কোন আইটেম বিয়োজন করাকে POP বলা হয়।

১৭। LIFO এবং FIFO এর পূর্ণনাম লেখ।

LIFO : Last In First Out.

FIFO : First In First Out.

১৮। STACK এ Overflow এবং Underflow বলতে কী বুঝায়।

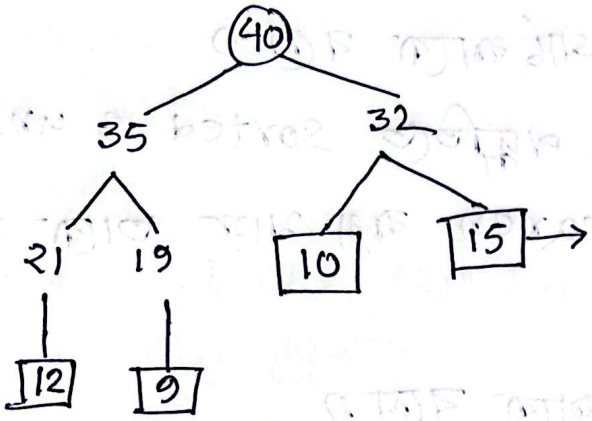
Overflow :- STACK এ নতুন Data সংযোজন করার জন্য যদি পর্যাপ্ত খালি যায়গা না থাকে সে অবস্থাকে Overflow বলা হয়।

Underflow :- STACK এ বিয়োজন করার মতো Data না থাকলে অথবা STACK খালি থাকলে সে অবস্থাকে Underflow বলে।

CamScanner

১৯। prefix Notation কী?
যে Notation পদ্ধতিতে operator-টি operand এর পূর্বে বসে, তাকে prefix Notation বলে।
২০। postfix Notation কী?
যে Notation পদ্ধতিতে operator-টি operand এর পরে বসে, তাকে postfix Notation বলে।
২১। Queue বলতে কী বুঝায়?
Data সমূহকে আরি বদ্ধ ভাবে সাজানোকে Queue বলে।
২২। De-Queue কাকে বলে?
De-Queue হলো একটির নের লিনিয়ার লিস্ট, যেখানে উপাদানগুলো তালিকার শেষে বা প্রথমে যুক্ত বা বিযুক্ত হতে পারে মাঝে নয়,
২৩। priority Queue কাকে বলে?
priority Queue হচ্ছে কিছু সংগৃহীত Data উপাদান যাদের প্রত্যেকটির জন্য priority নির্দিষ্ট করা থাকে,
২৪। লিংকড লিস্ট কাকে বলে?
Dynamic memory অ্যালোকেশন পদ্ধতিতে program ব্যবহৃত Data এর মাঝে একটি সম্পর্ক স্থাপন করা হয়, Data এর এরূপ সংযুক্তি কে Link List বলে।
২৫। সকল leaf-ই Node কিন্তু সকল Node, Leaf নয় কেন।

Leaf Node বলতে আপ আমরা ঐ সকল নোড গুলোকে বুঝি যার next pointer field হলো Null. প্রতিটি Leaf Node এর parent-Node থাকে, যা Tree-এর Root পর্যন্ত বিস্তৃত।



২৫। Tree বলতে কী বুঝায়?

যে Data Structure এর মাঝে Hierarchical সম্পর্কের সুস্পষ্ট প্রতিফলন ঘটে তাকে Tree বলে।

২৭। Tree Traversal বলতে কী বুঝায়?

যে পদ্ধতিতে কোন একটি ~~Node~~ Tree এর সকল নোড গুলো ডিজিট-বার তাদের মান print করা হয় তাকে Tree Traversal বলে।

২৮। স্প্যানিং ট্রি বলতে কি বুঝায়?

একটি গ্রাফের সাব সেটকে স্প্যানিং ট্রি বলে।

২৯। সার্চিং বলতে কী বুঝানো হয়?

অ্যারে হতে নির্দিষ্ট উপাদানের মান খুজে বের করাকে সার্চিং বলে।

৩০। বাইনারি সার্চিং কাকে বলে?

যে searching পদ্ধতিতে শুধু মাত্র Sorted data search করা যায় তাকে Binary search বলে।

৩১। লিনিয়ার সার্চ কাকে বলে?

যে searching পদ্ধতিতে sorted ও unsorted উভয় প্রকার data search করা যায় তাকে লিনিয়ার সার্চ বলে।

৩২। Sorting কাকে বলে?

Data কে ক্রমানুসারে সাজানোর পদ্ধতিকে Sorting বলে।

৩৩। Swapping বলতে কী বুঝায়?

যে পদ্ধতি অবলম্বন করে দুইটি Data পরস্পর স্থান বিনিময় করে তাকে swapping বলে।

৩৪। Data Type গুলো উল্লেখ কর।

Data Type দুই টি।

(i) Constant Data.

(ii) Variable data.

৩৫। (x+y) x z এর postfix notation লেখ।

(x+y) * z

⇒ (x+y) z *

⇒ xy z + *

৩৫। LOC নির্ণয়ের সূত্রটি লেখ।

LOC (LA[K]) = Base (LA) + W (K - LB) - - - -

৩৭। Complexity নির্ণয়ের তিনটি কারণ কী কী?

(I) Worst case.
(II) Average case.
(III) Best case.

৩৮। দুই-টি করে লিনিয়ার এবং নন লিনিয়ার Data Structure এর নাম লেখ।

লিনিয়ার :-

(I) STACK.
(II) Array.

নন লিনিয়ার :-

(I) Tree.
(II) Graph.

CamScanner

"ডাটা স্ট্রাকচার"

সংক্ষিপ্ত :-

১। ডাটা ও ইনফরমেশানের মধ্যে পার্থক্য লেখ?

ডাটা	ইনফরমেশান
(i) ডাটা শব্দের অর্থ উপাত্ত। কোন কিছুর মান বা ভ্যালুকে ডাটা বলে।	(i) ডাটাকে প্রক্রিয়াজাত করার পর যে অর্থপূর্ন মান পাওয়া যায় তাকে ইনফরমেশান বলে।
(ii) ডাটা নিজে অর্থপূর্ন নয়।	(ii) ইনফরমেশান অবশ্যই অর্থপূর্ন হয়।
(iii) কোন ছবি, তথ্য, বা মান হলো ডাটা।	(iii) ডাটাকে অ্যানালাইসিস করে যে মান পাওয়া যায় তা হলো ইনফরমেশান।
(iv) ডাটা হলো জ্ঞানের সর্বনিম্ন স্তর।	(v) ইনফরমেশান জ্ঞানের দ্বিতীয় স্তর।

২। ডাটাস্ট্রাকচার এর অপারেশান গুলো বর্ননা কর।

Traversing :- কোন Record কে মাত্র একবার Access করার পদ্ধতিকে Traversing অপারেশান বলে।

Searching :- কোন নির্দিষ্ট কী-ভ্যালু এর মাধ্যমে Record থেকে কোন Location খুজে বের করার পর পদ্ধতিকে Searching বলে।

Inserting :- Data Structure এর পুরাতন কোন রেকোর্ডের সাথে নতুন মান সংযুক্ত করার প্রক্রিয়াকে Inserting বলে।

Deleting :- Data Structure এর Record থেকে কোন আইটেম কে বাদ দেওয়াকে Deleting বলে।

Shorting :- Data Structure এর উপাদান সমূহকে মানের ক্রমানুসারে সাজানোকে Shorting বলে।

CamScanner

Merging :- Data structure এর দুই বা ততোধিক Record কে একত্রিত করাকে Merging বলে।

৩। Tree এবং Graph এর মধ্যে পার্থক্য।

Tree	Graph
(I) একটি Tree হলো NODE এবং EDGE এর একটি সেট।	(I) একটি Graph হলো Vertice/Node এবং EDGE এর একটি সেট।
(II) Tree তে একটি Unic Node থাকে যা Root নামে পরিচিত	(II) Graph এ কোন Unic node নেই যা Root নামে পরিচিত
(III) Tree কোন Loop গঠন করতে পারে না।	(III) Graph Loop গঠন করতে পারে।
(IV) একটি Tree তে একাধিক Child Node থাকতে পারে।	(IV) গ্রাফে কোন Child Node থাকে না।

৪। ডাটাস্ট্রাকচারের প্রকারভেদ লেখ।

(I) প্রিমিটিভ ডাটা স্ট্রাকচার।
(II) নন প্রিমিটিভ ডাটা স্ট্রাকচার।

* প্রিমিটিভ ডাটা স্ট্রাকচার
(I) ইন্টিজার।
(II) ফ্লোটিং।
(III) বুলিয়ান।
(IV) ক্যারেক্টার।

* নন প্রিমিটিভ ডাটা স্ট্রাকচার
(I) লিনিয়ার ডাটা স্ট্রাকচার -
(II) নন - লিনিয়ার ডাটা স্ট্রাকচার -

* লিনিয়ার ডাটা স্ট্রাকচার -
(I) অ্যারে।
(II) লিংকড লিস্ট।
(III) স্ট্যাক।
(IV) কিউ।

* নন-লিনিয়ার ডাটা স্ট্রাকচার -
(I) Tree
(II) Graph.

৫। Algorithm এর বৈশিষ্ট্য সমূহ লেখ –

(i) Algorithm এর নির্দিষ্ট সংখ্যক Input থাকবে।
(ii) Algorithm এর নির্দিষ্ট সংখ্যক ধাপ/step থাকবে।
(iii) Algorithm সহজ, সংক্ষিপ্ত ও নির্ভুল হবে।
(iv) Algorithm নির্বাহের সময়কাল নির্দিষ্ট সীমায় থাকবে।

৬। $Ax^2+Bx+C=0$ সমীকরণের মূল নির্ণয়: অ্যালগরিদম লেখ –

প্রদত্ত সমীকরন – $Ax^2+Bx+C=0$

নিশ্চায়ক $D = b^2-4ac$। যদি $(D>0)$ হয় তাহলে
$x_1 = -b+\sqrt{}/2a$ এবং $x_2 = -b-\sqrt{}/2a$ হবে।
আর যদি $(D=0)$ হয় তাহলে $x = -b/2a$ হবে। অন্যথায় বাস্তব মূল পাওয়া যাবে না।

Algorithm:

Step 1 : Start
Step 2 : set (a, b, c)
Step 3 : $D = b^2-4ac$
Step 4 : if (D>0) : then :
$x_1 = -b+\sqrt{}/2a$ and
$x_2 = -b-\sqrt{}/2a$
Write x_1 and x_2
Step 5 : else if (D=0) : then:
$x = -b/2a$
Write x =
Step 6 : else :
Write "Root are Imaginary"
Step 7 : Exit.

৭। কোন সংখ্যার ফ্যাক্টরিয়াল বের করতে অ্যালগরিদম লিখ –

Algorithm:

Step 1 :- Start
Step 2 :- Set N = 1
Step 3 :- If N = 0 : Then:
Fact = 1
Step 4 : Fact = 1
Step 5 : Reapeat For K = 1 to N.
Step 6 : Set fact = fact * K
Write fact
Step 7 : Exit.

৮। ফিবোনাক্কি সিরিজ নির্ণয় অ্যালগরিদম –

Algorithm :

Step 1 : Start
Step 2 : Set (a, b, c, n, i)
Step 3 : Set a = 0, b = 1, i = 2
Step 4 :- print a and b
Step 5 :- if $i \leq n$; then
c = a + b;
print c
a = b , b = c
i = i + 1
Step 6 : Exit.

৯। অ্যালগরিদমের কমপ্লেক্সিটি নির্ণয় করা কী? বর্ণনা কর।

Data structure ও বিভিন্ন ধরনের অপারেশন রয়েছে। যেমন:- Traversing, Searching, Shorting, Deleting, inserting ও Marging. এই operation গুলোর মাধ্যমে Data process হয়ে থাকে। Data processing এর জন্য প্রয়োজন Data এর গুলনা, আর যত বেশি Data গুলনা করা হবে তত বেশি সময় ও প্রয়োজন হবে। কোন Algorithm এর Running time কে Complexity function দ্বারা প্রকাশ করা হয়।

Algorithm এর complexity function $f(n)$ কে কয়েকটি Notetion দ্বারা প্রকাশ করা হয়। নিচে Notetion গুলো দেওয়া হলো:

Big-O-Notation/ Big-oh-Notation :-

The function of this Notation is $f(n) = O(g(n))$;

Big Omega notation :-

The function of this Notation is $f(n) = \Omega(g(n))$;

Theta notation :-

The function of this Notation is $f(n) = \theta(g(n))$;

Little 'oh' notetion :-

The function of this Notation is $f(n) = o(g(n))$;

Little 'omega Notation :-

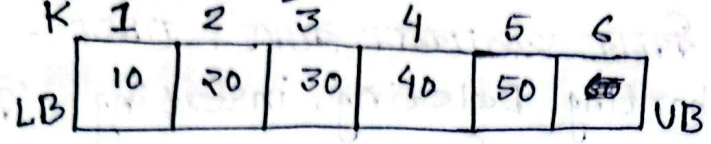
The function of this Notation is $f(n) = \omega(g(n))$;

কিছু পরিচিত Algorithm এর complexity দেওয়া হলো:

a. Binary Search : $O(\log n)$ b. Meargeshort : $O(n \log n)$

c. Linear Search : $O(n)$ d. bubble short : $O(n^2)$

১০। লিনিয়ার অ্যারেতে ডাটা সংযোজনের অ্যালগরিদম –



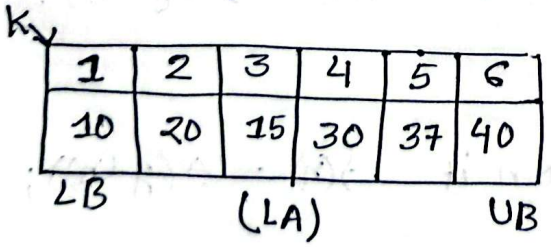
এখানে, N মোট উপাদান সংখ্যা, LA লিনিয়ার অ্যারে, LB লোয়ার বাউন্ড, UB আপার বাউন্ড। J ভেরিয়েবল নাম। এবং K = LOC.

Algorithm:

Step 1 :- Set J = N
Step 2 : Repeat Step 3 and 4 while : J <= K :
Step 3 : LA[J+1] = LA[J]
Step 4 : Set : J = J-1
Step 5 : Set LA[K] = Item.
Step 6 : Set N = N+1.
Step 7 :- Exit.

এই অ্যালগরিদম এর মাধ্যমে লিনিয়ার অ্যারেতে একটি ডাটা সংযুক্ত করা হয়।

১১। লিনিয়ার অ্যারেতে ডাটা বিয়োজন করার অ্যালগরিদম:



N = 6
বিয়োজিত আইটেম = 30
∴ LA[4] = 30

এখানে, K হলো লিনিয়ার অ্যারের লোকেশন নম্বর, N হলো উপাদান সংখ্যা। LA লিনিয়ার Array. এখানথেকে LA[4] উপাদান টি বাদ দিতে হবে। এই Algorithm এর মাধ্যমে আমরা উপাদান সমূহকে বাদ দিতে পারি।

Algorithm:

Step 1: Set LA[K] = Item.

Step 2:- Repeat ~~step 3 and 4~~
for J = K to N - 1
Set LA[J+1] = J

Step 3 : Set N = N - 1

Step 4 : Exit.

এই অ্যালগরিদমের মাধ্যমে LA থেকে একটি ডাটা বাদ দেওয়া হয়।

১৬। কোন Array এর N সংখ্যক Data থেকে বৃহত্তম Data এর মান ও LOC বের করার অ্যালগরিদম :

Algorithm :-

K	1	2	3	4	5	6	7	8
Data	10	12	14	20	16	18	30	22

→ (MAX)

Step 1:- Set K = 1, LOC = 1, MAX = Data[K]

Step 2: Repeat step 2 and 3 while N >= K :

Step 3: IF ~~MAX < Data[K]~~
MAX < Data[K] then:

LOC = [K]
MAX = Data[K]

K = LOC
Data[K] = ২P

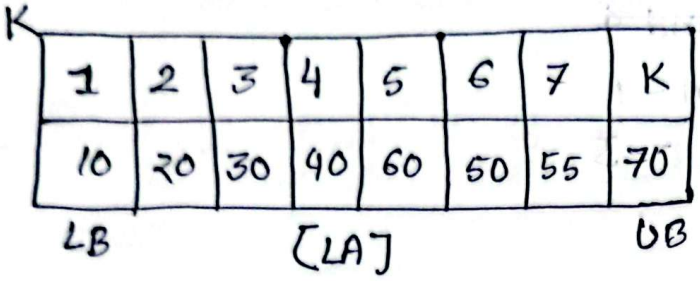
Step 4 : Set K = K+1

Step 5 : Write LOC, MAX

Step 6 : Exit.

1	2	3	4	5	6	7	K
10	12	14	16	18	8	6	20

০২। কোন Array এর N সংখ্যক Data এর মধ্যে বৃহত্তম Data ও তার Loc নির্ণয় Algorithm.

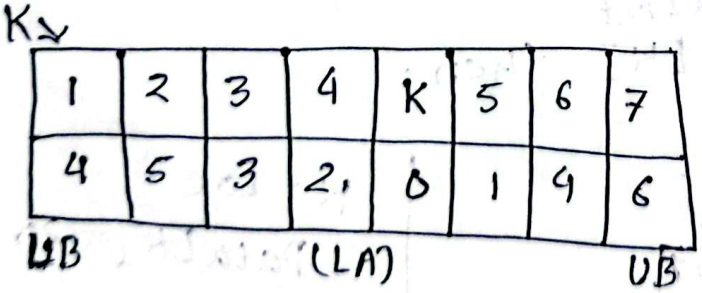


এখানে, K = Location
LB = Lower Bound
UB = Upper Bound
LA = Linear Array

Algorithm :-

Step 1 :- Set K = 1, LOC = 1, and Max = Data[K]
Step 2 :- Repeat Step 2 and 3 While K <= N :
Step 3 : IF Max < Data[K] then :
Set LOC = K, and Max = Data K
Step 4 : Set K = K+1
Step 5 : Write LOC, Max
Step 6 : End.

০৩। কোন Array এর N সংখ্যক Data থেকে ক্ষুদ্রতম Data ও তার LOC নির্ণয় Algorithm.



Algorithm.

Step 1 : Set K = 1, LOC = 1 and Data MIN = Data[K]
Step 2 : Repeat step 2 and 3 While K <= 3 :
Step 3 : If MIN > Data[K] Then : LOC = K, MIN = Data[K]
Step 4 : K = K+1

Step 5: print LOC, MIN.
Step 6: End.

১৪। Matrix multiplication Algorithm লেখ।

এখানে A হলো: MXP মাত্রার একটি Array এবং B হলো: PXN মাত্রার আরেকটি অ্যারে। A ও B এর গুনফল C নামক Array তে সংরক্ষন করা হয়।

Algorithm:
Step 1: & Repeat Step 2 to 4 for I = 1 to M:
Step 2: Repeat Step 3 to 4 for J = 1 to N:
Step 3: Set C[I,J] = 0
Step 4: Repeat for K = 1 to P:
Set C[I,J] = C[I,J] + A[I,K] * B[K,J]
Step 5: Exit.

১৫। Linear Array Traversing Algorithm. লেখ;

এখানে, LB লোয়ার বাউন্ড, UB আপার বাউন্ড, এবং Algorithm process এর মাধ্যমে LA Traverse করা হয়।

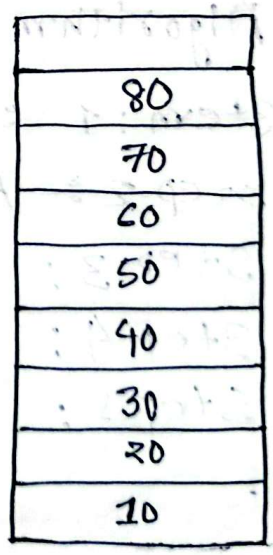
Algorithm:
Stem: 1 :- Set K: = LB.
Step 2:- Repeat step 3 and 4 while K ≤ UB.
Step 3: Apply process on LA[K].
Step 4: Set K = K+1
Step 5: Exit.

১৬/* পুশ এবং পপ অপারেশন এর মধ্যে পার্থক্য-

পুশ	পপ
(i) স্ট্যাক ডাটাস্ট্রাকচারে নতুন উপাদান সংযুক্ত করার প্রক্রিয়াকে পুশ অপারেশন বলে।	(i) স্ট্যাক ডাটাস্ট্রাকচারে কোন ডাটা অপসারণ করার জন্য যে অপারেশন করা হয় তাকে পপ অপারেশন বলে।
(ii) পুশ অপারেশনের ক্ষেত্রে স্ট্যাকের শীর্ষে ডাটা সংযোজন করানো হয়।	(ii) পপ অপারেশনের ক্ষেত্রে স্ট্যাকের শীর্ষ থেকেই ডাটা বিয়োজন করা হয়।
(iii) স্ট্যাকে ডাটা সংযোজনের জন্যে পুশ অপারেশন করা হয়।	(iii) স্ট্যাকের ডাটা পুনরুদ্ধার করার জন্যে পপ অপারেশন
(iv) পুশ অপারেশনে ওভারফ্লো সম্পন্ন হয়।	(iv) পপ অপারেশনে আন্ডারফ্লো সম্পন্ন হয়।

১৭। Stack এ Data Inserting Algorithm লেখ।
অথবা Stack এ Data push করার Algorithm লেখ।

যেখানে, MAXSTK = 8, and TOP = 80



Algorithm :
Step 1: Start.
Step 2: If MAXSTK = TOP Then :
print "Overflow" and go to Next step.
Step 3: TOP = TOP + 1
Step 4: Set Stack[TOP] = Item
Return.
Step 5: End.

১৮। Stack এ Data বিয়োজন / Deleting / POP অপারেশনের Algorithm লেখ।

Algorithm:

Step 1: Start

Step 2: If Top = Null : then: print "Underflow"
else go to next Step.

Step 3: Top = Top - 1;

Step 4: Set stack[Top] = Item and Return

Step 5: End.

১৯। স্ট্যাক এবং কিউ এর মধ্যে পার্থক্য লেখ-

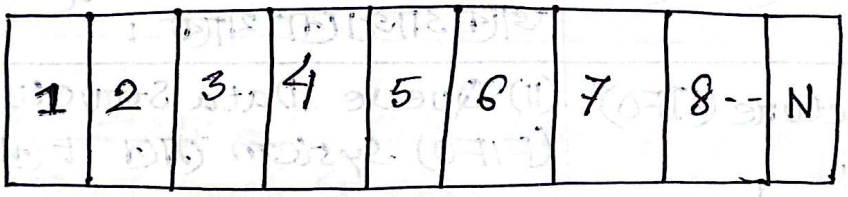
Stack	Queue
(i) Stack এমন এক ধরনের ডাটা সাংগঠন যেখানে ডাটাসমূহ স্তূপের মতো সাজানো থাকে।	(i) Queue এমন এক ধরনের ডাটা সাংগঠন যেখানে ডাটাসমূহ ধারাবাহিক ভাবে সাজানো থাকে।
(ii) Stack Data Structure (LIFO) System মেনে চলে।	(ii) Queue Data Structure (FIFO) System মেনে চলে।
(iii) Stack এ Data সংযোজন করাকে PUSH এবং বিয়োজন করাকে POP অপারেশন বলে।	(iii) Queue এ Data সংযোজন করাকে ENQUEUE এবং বিয়োজন করাকে DEQUEUE বলে।
(iv) Stack এ একটি নির্দেশকের প্রয়োজন।	(iv) Queue তে দুইটি নির্দেশকের প্রয়োজন।

২০। LIFO এবং FIFO System এর পার্থক্য লেখ।

(pto) →

FIFO	LIFO
(I) FIFO হলো First in First Out System.	(I) LIFO হলো Last in First Out System.
(II) Queue Data Structure কে FIFO System মেনে চলা হয়।	(II) Stack Data Structure কে LIFO System মেনে চলা হয়।
(III) বাস কাউন্টারের সামনে যাত্রীরা FIFO System মেনে চলে।	(III) টেবিলে বই সাজানোর ক্ষেত্রে LIFO নিয়ম মান হয়
(IV) FIFO System কে Data সমূহ এক প্রান্ত থেকে সংযোজন এবং অন্য প্রান্ত থেকে বিয়োজন হয়ে থাকে।	(IV) LIFO System কে Data সমূহ যে কোন এক প্রান্ত থেকে সংযোজন এবং বিয়োজন হয়ে থাকে

২১। Queue কে Data সংযোজন / EnQueue Algorithm লেখ-



N = Data এর পরিমান

Front = 1 Rear = 8

এখানে Queue এর প্রথম Data ~~Rear~~ Front = 1 এবং শেষ Data Rear = 8। Queue তে Data সংযোজনের ক্ষেত্রে Rear থেকে সংযোজন হবে যেহেতু Queue FIFO System মেনে চলে।

Algorithm:

S1 :- Start

S2 :- IF front = 1 and Rear = N Then:
Print "Overflow" and else go to step 3

Step 3:- IF Front = 1 and Rear = 1 ~~এবং~~ ~~Then~~ :

And Rear < = N then :

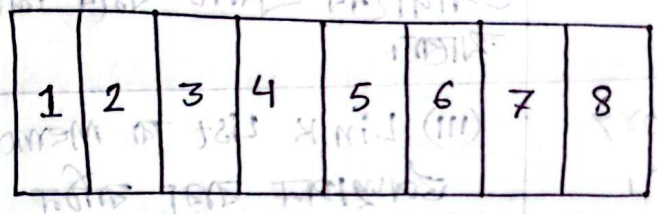
Set Rear = Rear + 1 and return.

Step 4: set Queue [Rear] = Item .

and return.

Step 5: Exit.

২২। Queue ৭ Data ডিলিট / DEQUEUE Algorithm.



Algorithm:

Step 1: Start.

Step 2: If Front = Null then:

Print "Underflow" and go to Step - 3

Step-3: If Front = Rear then :

Set Front = Null and Rear = Null.

else if Front = 1 and Rear = N. Then:

Set ~~Rear = Rear~~ FRONT = Front - 1

else::

Front = Front + 1

Set Front [Queue] = Item.

Step - 4 : Return

Step - 5 : Exit.

২৩। Array এবং Link list এর মধ্যে পার্থক্য :-

Array	Link List
(i) ডাটাসমূহকে পাশাপাশি শ্রেনীবদ্ধ ভাবে সাজানোকে Array বলে।	(ii) Dynamic Memory allocation পদ্ধতিতে program এ ব্যবহৃত Data সমূহকে সংযুক্ত করা হয়, এরূপ সংযুক্তিকে Link List বলে।
(ii) Array তে Data উপাদান পাশাপাশি সজ্জিত থাকে।	(ii) Link List এ Data সমূহ ছড়ানো ছিটানো থাকে। প্রতিটি উপাদানের একটি করে লিংক থাকে।
(iii) Array তে data কে memory তে উপস্থাপন করা সহজ।	(iii) Link List কে Memory তে উপস্থাপন করা কঠিন

২৪। গারবেজ কালেকশন, Overflow, এবং Underflow সম্পর্কে বর্ননা কর।

গারবেজ কালেকশন :- Garbage বলতে অপ্রয়োজনীয় অথচ তথ্যকে বুঝায়। কোনো লিস্টের যে সমস্ত নোডে এ জাতীয় তথ্য সঞ্চিত থাকে সে সকল নোডগুলোকে অপসারন পূর্বক Free Memory Space কে পুনরায় ব্যাবহার করা যেতে পারে।

Overflow :- কোন কোন সময় Data Structure এ নতুন Data সংযোজনের প্রয়োজন হয়। কিন্তু Data Structure এ যদি Data সংযোজনের জন্য প্রয়োজনী স্পেস না থাকে তখন সে অবস্থাকে Overflow বলা হয়, আর program এ এরূপ অবস্থায় Overflow print করে।

Underflow :- কোন কোন সময় Data Structure এর Store কৃত Data গুলি বিয়োজনের প্রয়োজন হয়। Data Structure এ যদি বিয়োজন করার মতো কোন Data

না থাকে তাহলে ঐ অবস্থাকে Underflow বলে। ঐ সময় program এ underflow print হবে।

২৫। Link List এর সুবিধা বা অসুবিধা বর্ননা কর।

সুবিধা:-

(i) Data এর সাইজ পূর্বে নির্ধারনের প্রয়োজন হয় না।

(ii) যে কোন সংখ্যক নতুন Data সংযোগ করা যায়।

(iii) List এর সাইজ সুবিধা মতো বৃদ্ধি করা যায়।

(iv) মেমরি অপচয় হ্রাস করা যায়।

অসুবিধা:-

(i) Link List এ Data সংযোজনের জন্যে লিংক স্পেস এবং এই স্পেসের জন্য অতিরিক্ত Data এর প্রয়োজন হয়।

(ii) Random access প্রক্রিয়ায় এটি উপযুক্ত নয়।

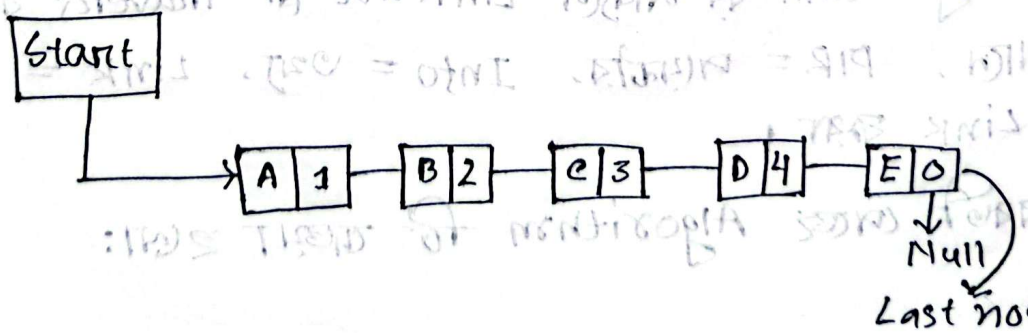
২৬। Memory তে কিভাবে লিংকড লিস্ট কে উপস্থাপন করা যায়।

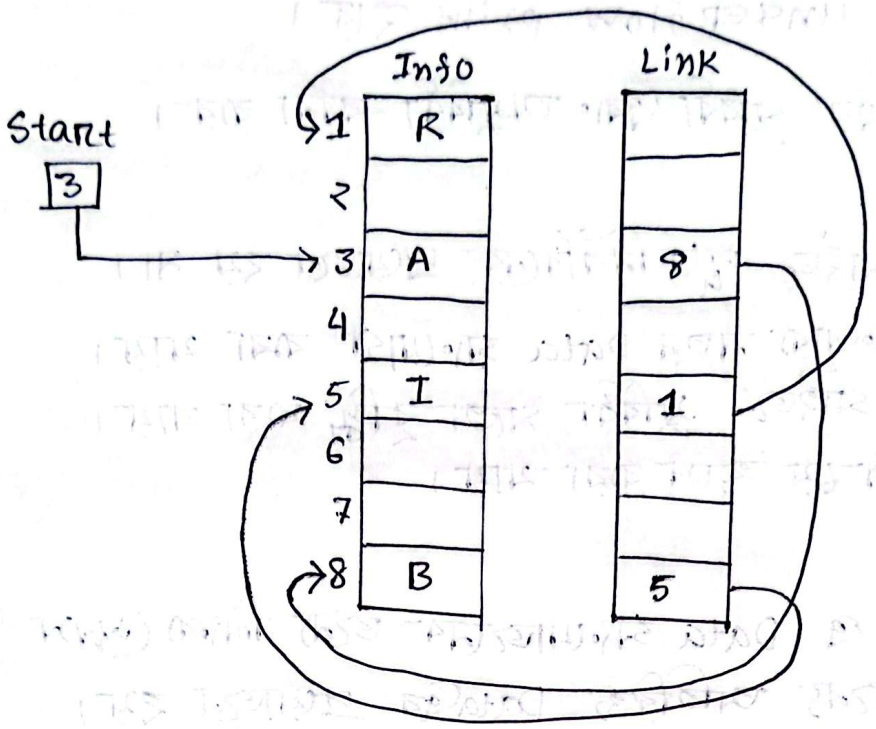
Memory তে লিঙ্কড লিস্ট দুইটি Array এর মাধ্যমে সজ্জিত থাকে, একটি Info এবং অন্যটি Link.

(i) Info (k) → List এর information part নির্দেশ করে।

(ii) Link (k) → List এর pointer বা পরবর্তি নোডকে নির্দেশ করে।

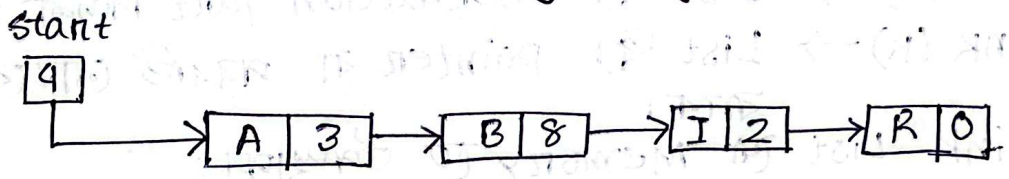
(iii) Link List কে Memory তে উপস্থাপন:





Start = 3, Info(3) = A, It is a first character.
LINK(3) = 8, Info(8) = B, It is Second character.
Link (8) = 5, Info (5) = I, It is Third character.
Link (5) = 1, Info (1) = R, It is 4th character.

২৭। Link List এর Traversing Algorithm লেখ।



এই Algorithm এর মাধ্যমে Link List কে Traverse করা হয়। এখানে, PTR = পয়েন্টার, Info = তথ্য, Link = পরবর্তী node কে Link করা।

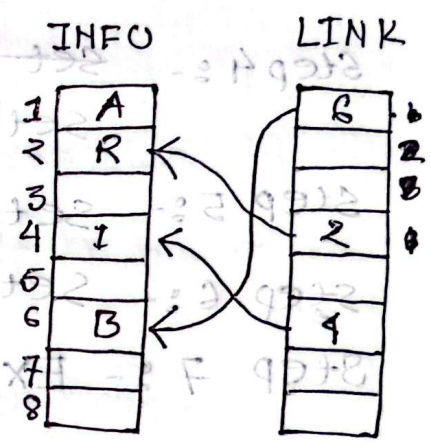
পরবর্তী পেজে Algorithm টি দেওয়া হলো:

Algorithm:

Step 1: Start
Step 2: Set START = PTR
Step 3: Repeat step 4 and 5 while PTR ≠ NULL
Step 4: Apply process PTR[INFO]
Step 5: Set PTR = LINK[PTR]
Step 6: Exit.

২৮। লিঙ্কড লিস্টে ডেটা সংযোজনের অ্যালগরিদম লেখ।

এখানে, START = শুরু, AVAIL হলো খালি নোডের পরিমান, INFO = তথ্য, LINK = সংযুক্ত করা, এই Algorithm এর সাহায্যে লিঙ্কড লিস্টে নতুন Data সংযোজন করা হয়।



Algorithm:-

Step 1 :- Start
Step 2 :- If AVAIL = NULL: Then: ~~go to~~ Next
print "overflow" and Exit.
Step 3 :- Set NEW=[AVAIL] and
Set AVAIL = NEW [AVAIL]

২৫। Link List এ Data অন্তর্ভুক্তের Algorithm লেখ।

Algorithm :- INSLOC [INFO, NEW, AVAIL, LINK, LOC, START.]

Step 1 :- Start.
Step 2 :- If AVAIL = NULL Then:
Print "overflow" and Exit.
Step 3 :- Set INFO[NEW] = Item.
Step 4 :- If LOC = NULL then:
Set LINK[NEW] = START
and START := NEW
Step 5 :- else: LINK[NEW] = LINK[LOC]
and LINK[LOC] := NEW
Step 6 :- EXIT.

২৬। Link List এ Data বিয়োজনের Algorithm.

Algorithm : DELLOC [LINK, LOC, LOCP, AVAIL, START UNDERFLOW,]

Start : IF LOC = NULL Then:
print "Underflow"

Step 1 :- Start.
Step 2 :- If LOC = NULL then:
print "Underflow" and EXIT
Step 3 :- If LOCP = NULL then:
START := LINK[LOC]
Step 4 : else :
LINK[LOCP] = LINK[LOC]

Step5 :- LINK[LOC] = AVAIL

AVAIL = ~~LINK~~ [LOC]

Step6: Exit.

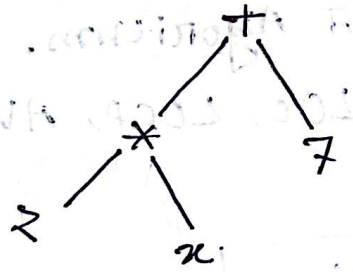
৩০। Tree ট্রাভার্স বলতে কী বুঝায়? কটি কত প্রকার-কি কি কী?

যে পদ্ধতিতে একটি Tree এর সকল নোডগুলো এমন ভাবে তাদের মানগুলোকে প্রিন্ট করা হয় তাকে Tree ট্রাভার্সাল বলে।

তিনভাবে Tree ট্রাভার্সাল করা যায়-

(i) In-Order Traversal.
(ii) pre-Order Traversal.
(iii) post-Order Traversal.

৪০। 2x+7 কে Tree আকারে প্রকাশ কর-

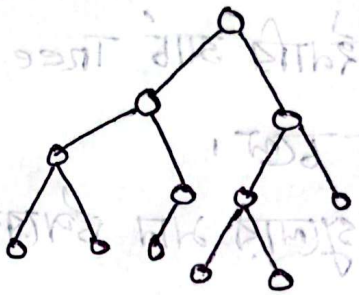


৩১।

৪৪। Tree কাকে বলে বিভিন্ন প্রকার Tree বর্ননা কর।

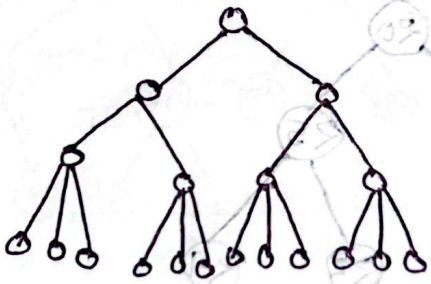
যে Data structure এ Hierarchical সম্পর্কের মধ্যে প্রতিফলন ঘটে তাদেরকে Tree Data Structure বলে। Data Structure এ বিভিন্ন প্রকার Tree Structure ব্যবহৃত হয়ে আসছে। নিচে তাদের বর্ননা দেওয়া হলো :-

(i) M-array Tree: যে সকল Tree তে M সংখ্যক বা তার কম সংখ্যক চাইল্ড থাকে তাদেরকে M-array Tree বলে:-

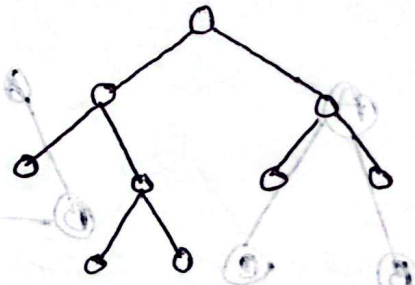


M-Array Tree

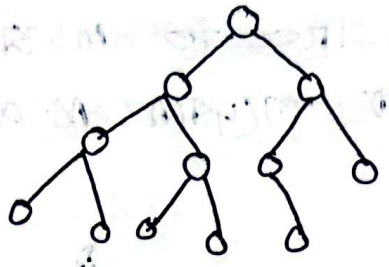
(ii) Full M-Array Tree :- যে সকল Tree তে exactly M সংখ্যক child থাকে তাকে Full-M-Array Tree বলে।



(iii) Binary Tree :- যে সকল Tree তে সর্বোচ্চ দুইটি child থাকে তাকে Binary Tree বলে।



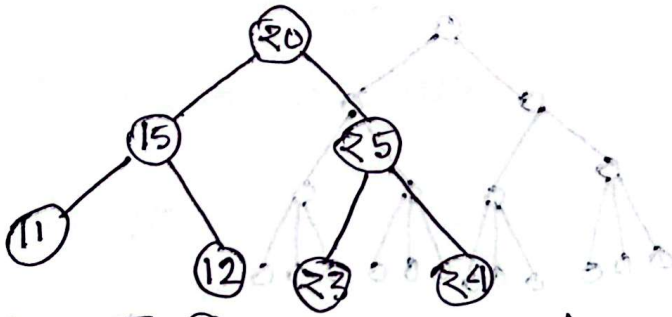
(iv) Complete Binary Tree: Complete বাইনারি Tree এমন এক ধরনের Tree যার Last Level ব্যতিত সকল Level এ N পরিপূর্ণ থাকবে।



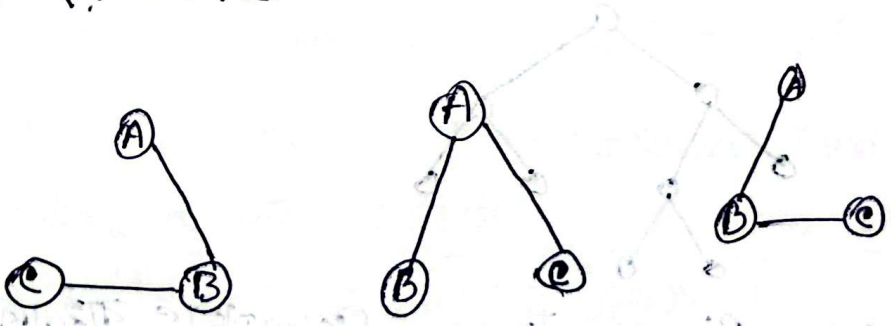
(v) বাইনারি সার্চ Tree :- বাইনারি সার্চ Tree যেমন এক-ধরনের Tree যা নিচের নিয়ম মেনে চলে।

(i) বাম দিকের সাব Tree গুলোর মান উপরের মানের তুলনায় ছোট বা সমান হবে।

(ii) ডান দিকের সাব Tree গুলোর মান উপরের মানের থেকে বড় বা সমান হবে।



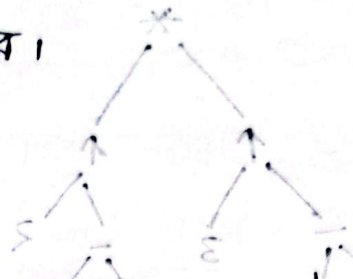
(vi) স্প্যানিং ট্রি :- একটি গ্রাফের সাব সেট কে spanning Tree বলে। যদি A, B, C তিনটি ভার্টিসেস নিয়ে একটি Tree তৈরি হয় তাহলে।



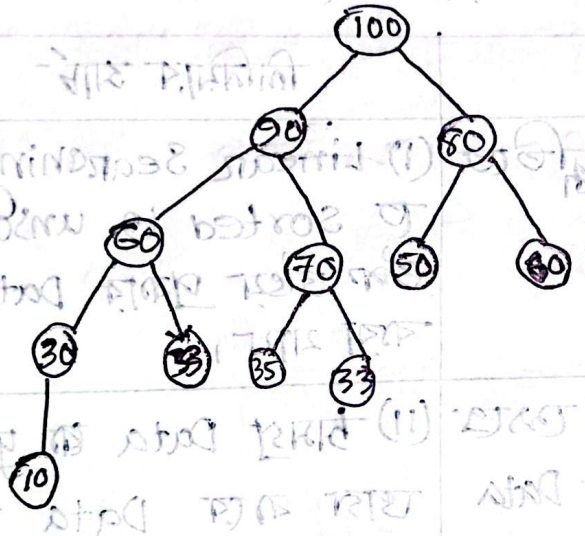
Heap Tree :- একটি Tree উপাদান কে তখনি Heap Tree বলা যাবে যখন বিভিন্ন নোড উপাদান সমূহকে মানের ক্রমানুসারে সাজানো থাকে। Heap Tree ২ প্রকার।

(I) MAX Heap Tree.

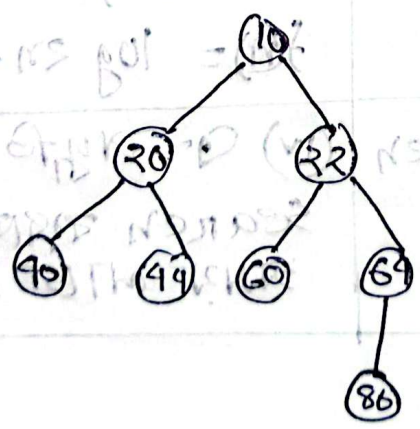
(II) MIN Heap Tree.



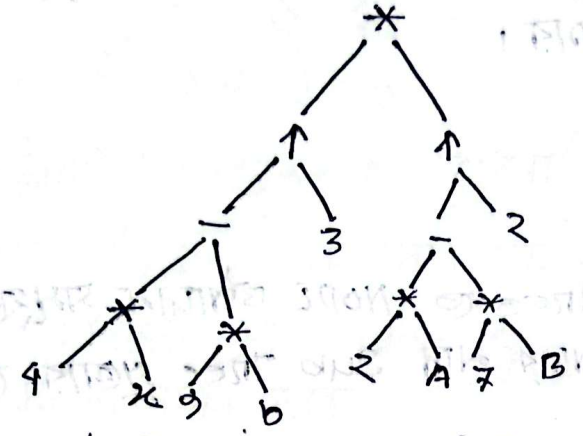
(I) MAX Heap Tree :- এ Tree তে NODE উপাদান সমূহকে মানের উর্ধ্ব ক্রমানুসারে সাজানো থাকে, উপরের মান Sub Tree গুলোর থেকে বড় হবে।



(II) MIN Heap Tree :- এ Tree তে Node উপাদান সমূহকে মানের নিম্ন ক্রমানুসারে সাজানো থাকে। উপরের মানের তুলনায় নিচের Tree এর মানগুলো বড় হবে।



৬২। $(4x - 9b)^3 (2A - 7B)^2$ কে Tree আকারে প্রকাশ করা।



৬৩। বাইনারি সার্চ এবং লিনিয়ার সার্চের মধ্যে পার্থক্য লেখ।

বাইনারি সার্চ-	লিনিয়ার সার্চ-
(i) বাইনারি সার্চিং শুধু পদ্ধতিতে শুধুমাত্র Shorted Data Search করা যায়।	(i) Linear Searching এ পদ্ধতি তে Sorted ও unsorted উপর উভয় প্রকার Data Search করা যায়।
(ii) Data Searching এর ক্ষেত্রে প্রথমে থেকে পর্যায়ক্রমে Data তুলনা করা শুরু হয়।	(ii) সমগ্র Data কে দুই ভাগে ভাগ করে Data তুলনা করা শুরু হয়।
(iii) Binary Search এর Complexity $f(n) = n+1$	(iv) Linear Searching এর Complexity $f(n) = \log_2 n + 1$
(iii) এ পদ্ধতিতে Data Search করতে সময় কম লাগে।	(iv) এ পদ্ধতি তে Data Search করতে বেশি সময় লাগে।

৩৪। বাইনারি সার্চের সীমাবদ্ধতা আলোচনা কর –

(i) বাইনারি সার্চ ক্রমানুযায়ী সাজানো থাকে তাই লিস্টের Lenght অনেক বড় হয়।

(ii) কোন নতুন Data সংযোজন করতে হলে Searching এর Key word বেড়ে যায় এবং সময় বেশি লাগে।

৩৫। Linear Searching Algorithm বর্ণনা কর।

এই Algorithm এর মাধ্যমে unsorted Data Search করা হয়। এখানে, I. Index সংখ্যা, n সকল Data এর সংখ্যা, এবং Item হলো যার LOC search করা হবে।

Algorithm:- Search সফল হলে Data এর LOC প্রিন্ট করবে অন্যথায় LOC = 0 প্রিন্ট করবে।

Step 1 :- Set I to 1.

Step 2:- If I > n : Then: Go to step 7 and Exit.

Step 3 :- If I = Item : Then: Go to Step 6 and Exit.

Step 4:- Set I = I+1

Step 5:- Go to step 2

Step 6 :- print element item found and index LOC=i. Go to step 8

Step 7 :- print the element item not found and index LOC = 0. Go to step 8

Step 8 : Exit.

CamScanner

৬৮। বাইনারি search অ্যালগরিদম লেখ।

Algorithm:- Binary Search [MID, BEG, END, VAL, LB, UB, START]

Step 1: Set BEG = LB, END = UB, AND POS = -1

Step 2: Repeat step 3 and 4 while BEG ≤ END

Step 3: set MID = (LB+UB)/2.

Step 4: If MID = VAL Then: Go to step 6

else If MID >= VAL then:

END = MID - 1

else

BEG = MID + 1

Step 5:- If POS = -1 then:

print value is not present in this Array

Step 6: Exit.

৬৭। বাবল sort অ্যালগরিদম লেখ।

Algorithm: Bubble Sort [Data, PTR]

Step 1:- Set K = 1 to N-1

Step 2: Set PTR = 1

Step 3: Repeat step 4 and 5 while PTR ≤ N-K

Step 4: If Data[PTR] < Dat[PTR+1] then.

Swapping Item. And Exit.

Step 5: set PTR = PTR+1.

Step 6: Exit.

CamScanner

06) Quick Sort Algorithm লেখো।

Algorithm : Quick Sort [BEG, END, Upper, & Lower, LOC]

Step 1 : set Top = NULL.

Step 2: If N>1 then:
set Top = Top+1, Lower[1]=1, Upper[1]=1

Step 3 : Repeat step 4 to 6 while Top ≠ NULL

Step 4 : set BEG = Lower[TOP], END = Upper[TOP]
Top = Top-1.

Step 5 : If BEG > LOC-1 Then:
set Top = Top-1, Lower[Top] = BEG, Upper[TOP] = LOC-1.

Step 6 : I BEG < LOC+1 then:
set TOP = TOP+1, Lower[TOP] = LOC+1,
Upper[TOP] = BEG.

Step 7 : Exit.