

প্রাথমিক ধারণা

1) Data structure কাকে বলে? Data Structure এর প্রকারভেদ

উদাহরণ সহ লিখ।

Logical and mathematical model কে Data Structure বলে।

Classification of Data Structure: ২ প্রকার-

(i) Primitive DS (Built-in DS) :

Integer, Float, Character, Double, Pointer, Boolean

(ii) Non-primitive DS (User Defined DS): ২ প্রকার-

(a) Linear DS:

-Static DS: Examples: Array

-Dynamic DS: Example: Stack, Queue, Linked list

(b) Non-Linear DS: Example: Tree, Graph, Set, Table

2) Data Structure এর প্রয়োজনীয়তা লিখ।

Processor speed, Data search, Multiple request

3) Data Structure এর সুবিধাগুলো লিখ।

Efficiency, Reusability, Abstraction

4) Linear Data structure / Data Structure এর Basic operation লিখ।

(i) Insertion (ii) Deletion (iii) Sorting (iv) Searching

(v) Traversing (vi) Merging

3) Algorithm কাকে বলে? Algorithm এর বৈশিষ্ট্য লিখ।

An Algorithm is a finite set of instructions or logic written in order, to accomplish a certain predefined task.

Properties: (i) Input (ii) Output (iii) Definiteness
(iv) Finiteness (v) Correctness

4) Algorithm Performance কিসের উপর নির্ভর করে?

Time Complexity: The amount of time needed by a program to run to the completion

Space Complexity: The amount of memory space required by an algorithm, during a course of its execution.

Array

1) Array কাকে বলে? সুবিধা ও অসুবিধাগুলো লিখ।

Array is a group of elements (data). All the elements are similar. It has contiguous memory location.

সুবিধা: (i) Easy to traverse (ii) Code optimization

(iii) Easy to sort data (iv) Random access

অসুবিধা: (i) Insertion and deletion কঠিন। (ii) Time consuming (iii) যেহেতু Array sized fixed সেহেতু Run time এ Memory অপচয় হওয়ার সম্ভাবনা থাকে। (iii) Run time এ ভেঁমরি কমানো ও বাড়ানো যায়না।

2) Array প্রকারভেদ: (i) One dimensional (1-D)

(ii) Two dimensional (2-D) (iii) Multi-dimensional

3) One dimensional Array Linear array কে represent দেখা?

`int Student[5];`

Addresses				
1008	1006	1004	1002	1000
4	3	2	1	0

Index

-> প্রথম index এর address কে Base Address বলে। 1000 হচ্ছে Base Address (BA)।

-> First index কে upper bound (UB) বলে। UB=0

-> Last index কে Lower bound (LB) বলে। LB=4

-> Array এর প্রতি cell বা index এ যত বাইট জায়গা দখল করে তাকে word size (w) বলে।

Linear array length বা (Number of elements) বের করার সূত্র:

$\text{length} = \text{UB} - \text{LB} + 1$

Linear array এর index এর address বের করার সূত্র:

$\text{LA}[\text{index}] = \text{BA} + \text{W}(\text{index} - \text{LB})$

4) Apple company 2007 সাল থেকে 2020 পর্যন্ত প্রত্যেক সালের বিক্রিত iPhone সংখ্যার Record Data array তে reserve আছে। উক্ত Array element সংখ্যা কত?

Solution: $\text{Length} = \text{UB} - \text{LB} + 1 = 2020 - 2007 + 1 = 4$ (Ans)

5) মনে করো Array গুলো RRR(4:35), Student(-5:12), and Momentum(24) হলে প্রতিটি অ্যারের উপাদান সংখ্যা বা length বের করো।

Solution:

$\text{length}(\text{RRR}) = \text{UB} - \text{LB} + 1 = 35 - 4 + 1 = 32$ (Ans)

$\text{length}(\text{Student}) = \text{UB} - \text{LB} + 1 = 12 - (-5) + 1 = 18$ (Ans)

$\text{length}(\text{Momentum}) = \text{UB} - \text{LB} + 1 = 24 - 1 + 1 = 24$ (Ans)

Note: যদি lower bound দেয়া না থাকলে LB=1 ধরে নিতে হবে।

5) মনে করো একটি Array Rusafa(5:50) যাহার BA=200 and word size 4 হলে Rusafa[15], Rusafa[65] এর Address বের করো।

Solution: Rusafa[15] এর ক্ষেত্রে,

Given, LB=5, W=4, index=15

$\text{LA}(\text{Rusafa}[\text{index}]) = \text{BA} + \text{W}(\text{index} - \text{LB})$

$\text{LA}(\text{Rusafa}[15]) = 200 + 4(15 - 5) = 240$ (Ans)

Rusafa[65] এর ক্ষেত্রে,

Rusafa[65] হচ্ছে Rusafa array এর উপাদান নয়। সুতরাং বের করা সম্ভব নয়। কারণ UB=50

6) Two dimensional array কে Memory তে Represent দেখা?

`int Rehan[3][3];`

Rehan[0][0]	Rehan[0][1]	Rehan[0][2]
Rehan[1][0]	Rehan[1][1]	Rehan[1][2]
Rehan[2][0]	Rehan[2][1]	Rehan[2][2]

৪) ধরো একটি Two-dimensional Array

CSE(2:8, -4:1) ঘোষণা করা আছে।

Two-dimensional Array এর উপাদান বা দৈর্ঘ্য বের কর।

Solution: Given, $UB_1=8, LB_1=2, UB_2=1, LB_2=-4$

We know, $Length=(UB_1-LB_1+1) \times (UB_2-LB_2+1)$

$$=(8-2+1) \times \{(1-(-4))+1\} \times \{(10-6)+1\} = 7 \times 6 = 42 \text{ (Ans)}$$

=> যদি Array[M][N] হয় —

Column-major order এর সূত্র :

$$Array_Name[J,K]=BA+W\{(J-1)+M(K-1)\}$$

Row-major order সূত্র:

$$Array_Name[J,K]=BA+W\{N(J-1)+(K-1)\}$$

৭) ৫০ জন ছাত্রের ৪ টি Class test result সংরক্ষণের জন্য একটি 25X4 2D Array তে Student নামক

Array তে সংরক্ষণ আছে। Student Array এর First

Address হল 200 এবং প্রতি memory cell এ 4

word ধারণ করে। Column major order এবং

Row major order এ সংরক্ষণ উপযোগী Array টিতে

Student[12,3] এর Address নির্ণয় কর।

Solution: Given,

$$M=25, N=4, J=12, K=3, BA=200, W=4$$

Column-major order এ,

$$Student[J,K]=BA+W\{(J-1)+M(K-1)\}$$

$$\text{বা, } Student[12,3]=200+4\{(12-1)+25(3-1)\}=444 \text{ (Ans)}$$

Row-major order-এ,

$$Student[J,K]=BA+W\{N(J-1)+(K-1)\}$$

$$Student[12,3]=200+4\{4(12-1)+(3-1)\}=384 \text{ (Ans)}$$

৭) Three dimensional Array কে Memory তে Represent দেখা?

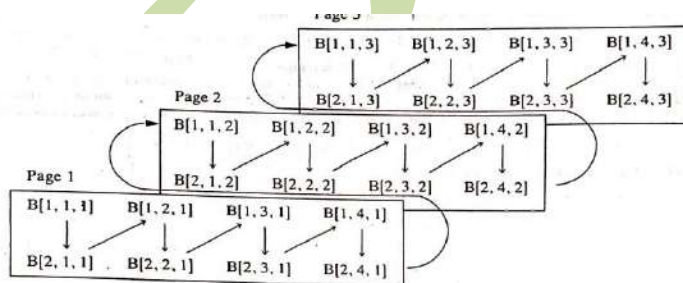


Fig. 4-11

B	Subscripts	B	Subscripts
	(1, 1, 1)		(1, 1, 1)
	(2, 1, 1)		(1, 1, 2)
	(1, 2, 1)		(1, 1, 3)
	(2, 2, 1)		(1, 2, 1)
	(1, 3, 1)		(1, 2, 2)
	⋮		⋮
	(1, 4, 3)		(2, 4, 2)
	(2, 4, 3)		(2, 4, 3)

(a) Column-major order.

(b) Row-major order.

৪) ধরো একটি Three-dimensional Array

DUET(2:8, -4:1, 6:10) ঘোষণা করা আছে।

Three dimensional Array এর উপাদান বা দৈর্ঘ্য বের কর।

Solution: Given

$$UB_1=8, LB_1=2, UB_2=1, LB_2=-4, UB_3=10, LB_3=6$$

We know,

$$Length=(UB_1-LB_1+1) \times (UB_2-LB_2+1) \times (UB_3-LB_3+1)$$

$$=(8-2+1) \times \{(1-(-4))+1\} \times \{(10-6)+1\} = 7 \times 6 \times 5 = 210 \text{ (Ans)}$$

৭) ধরো একটি Three-dimensional Array

DUET(2:8, -4:1, 6:10) ঘোষণা করা আছে।

DUET Array এর First Address হল 200 এবং প্রতি

memory cell এ 4 word ধারণ করে। Row major

order এ সংরক্ষণ উপযোগী Array টিতে DUET[5,-1,8]

এর Address নির্ণয় কর।

Solution: Given,

$$index_1=5, index_2=-1, index_3=8,$$

$$UB_1=8, LB_1=2, UB_2=1, LB_2=-4, UB_3=10, LB_3=6$$

আমরা জানি,

$$L_1=(UB_1-LB_1+1)=(8-2+1)=7,$$

$$L_2=(UB_2-LB_2+1)=\{(1-(-4))+1\}=6$$

$$L_3=(UB_3-LB_3+1)=10-6+1=5$$

$$E_1=index_1-LB_1=5-2=3$$

$$E_2=index_2-LB_2=-1-(-4)=3$$

$$E_3=index_3-LB_3=8-6=2$$

আবার,

$$Array_Name[index_1, index_2, index_3]$$

$$=BA+W\{(E_1 \times L_1 + E_2) \times L_3 + E_3\}$$

$$\text{বা, } DUET[5,-1,8]=200+4\{(3 \times 6 + 3) \times 5 + 2\}$$

$$=200+4(105+2)=200+4 \times 107=628 \text{ (Ans)}$$

৪) 2D array এর Application লিখ।

(i) Video displays (ii) Calendars (iii) Labeled lists

(iv) Graph (v) Image solve linear equation (vi)

Game boards (vii) Matrix processing etc

Pointer

Advantage of pointer :

(i) Program এর length কমে যায়। Complexity কমে যায়।

Memory Save করে।

(ii) increase Execution speed.

Disadvantages :

(i) Pointer handle করা কঠিন।

(ii) Run time available storage না থাকলে Program crash হতে পারে।

Stack

1) Stack কাকে বলে? উদাহরণ সহ লিখ।

Stack হচ্ছে Linear data structure যাহা ,LIFO (Last-In-First-Out) পদ্ধতিতে কাজ করে। যেই ডাটা প্রথমে Insert হবে সেই ডাটা সর্ব শেষে Delete হবে। একটি Top নামক variable ধরে কাজ আমরা কাজ করে থাকি।

Application:

- (i) Recursion (ii) parsing (iii) browsers
- (iv) Tree Traversals (v) Editors
- (vi) Expression evaluations

2) Stack এর Common operation লিখ।

- (i) Create () (ii) Push () (iii) pop () (iv) isFull ()
- (v) isEmpty () (vi) peek () (vii) Clear ()

3) What is the data structure used to perform recursion?

Ans: Stack

4) Stack এর PUSH and POP Algorithm লিখ।

PUSH	POP
1. If TOP=MAXSTK, Then :Print: Overflow and Return	1. If TOP=0, Then :Print: Underflow and Return
2. Set TOP:=TOP+1	2. Set ITEM:=STACK[TOP]
3. Set STACK[TOP]:=ITEM	3. Set TOP:=TOP-1
4. Return	4. Return

5) (i) $((x+y) \uparrow 2) + ((x-y)/3)$ postfix এ রূপান্তর কর।

(ii) $((x+y) \uparrow 2) + ((x-y)/3)$ prefix এ রূপান্তর কর

Ans: Self study

6) মান নির্ণয় কর।

calculate the value of following postfix expression using Stack
5,6,2,+,*,12,4/, -

Solⁿ: 5,6,2,+,*,12,4/, -

Scanned	Stack	Scanned	Stack
5	5	12	40,12
6	5,6	4	40,12,4
2	5,6,2	/	40,3
+	5,8	-	37
*	40)	

calculate the value of following postfix expression using Stack

(i) 12,7,3,-/,2,1,5,+,*,+ (ii) 5,3,+,2,1,8,4/,3,6,+,*, -

Solⁿ: (i) 12,7,3,-/,2,1,5,+,*,+

(ii) 5,3,+,2,1,8,4/,3,6,+,*, -

Scanned	Stack
12	12
7	12,7
3	12,7,3
-	12,4
/	3
2	3,2
1	3,2,1
5	3,2,1,5
+	3,2,6
*	3,12
+	15

Scanned	Stack
5	5
3	5,3
+	8
2	8,2
1	64
8	64,8
4	64,8,4
/	64,2
3	64,2,3
6	64,2,3,6
+	64,2,9
*	64,18
-	46

Infix to Postfix conversion using STACK:

(a) $(A-B) \uparrow (D/E)$ (c) $A+(B \uparrow C-(D/E \uparrow F)) \uparrow G \uparrow H$

(b) $(A+B \uparrow D) \uparrow (E-F) \uparrow G$

Solⁿ: (a) $((A-B) \uparrow (D/E))$

Solⁿ: (b) $((A+B \uparrow D) \uparrow (E-F) \uparrow G)$

Scanned	Stack	Postfix expression
((
A	((A
-	((-	A
B	((-	AB
)	(AB-
*	(*	AB-
((*(AB-
D	(*(AB-D
/	(*/	AB-D
E	(*/	AB-DE
)	(AB-DE/
)		AB-DE/*

Scanned	Stack	Postfix expression
((
A	((A
+	((+	A
B	((+	AB
↑	((↑	AB
D	((↑	ABD
)	(ABD↑
/	(/	ABD↑
(((ABD↑
E	((ABD↑E
-	((-	ABD↑E
F	((-	ABD↑EF
)	(ABD↑EF-
+	(+	ABD↑EF-
G	(+	ABD↑EF+G
)		ABD↑EF+G+

9) Given the following arithmetic expression in infix notation: $25/(9+3)+7*(13+10)$ Translate this expression into postfix notation and then evaluate it using stack by showing all necessary steps [Ans: 163.08]

10) নিচের রাশিটি Stack এর মাধ্যমে মান নির্ণয় কর।

$A+(B \uparrow C-(D/E \uparrow F)) \uparrow G \uparrow H$

Solⁿ: (c) $(A+(B^*C-(D/B)^F)^G)^H$

Scanned	Stack	Postfix expression
((
A	(A
+	(+	A
(((A
B	((AB
*	((*	AB
C	((*	ABC
-	((-	ABC*
((((-	ABC*
D	(((-	ABC*D
/	(((/	ABC*D
E	(((/	ABC*DE
↑	(((/↑	ABC*DE
F	(((/↑	ABC*DEF
)	(((-	ABC*DEF↑
*	(((*	ABC*DEF↑
G	(((*	ABC*DEF↑G
)	((ABC*DEF↑G*
+	((+	ABC*DEF↑G*+
H	((+	ABC*DEF↑G*+H
)		ABC*DEF↑G*+H↑

calculate the value of following Infix expression using Stack

$$(5+3)↑2-(8/4)^*(3+6)$$

Scanned	Stack	Postfix expression
((
(((
5	((5
+	((+	5
3	((+	5,3
)	((5,3,+
↑	((↑	5,3,+
2	((↑	5,3,+,2
-	((-	5,3,+,2,↑
((((-	5,3,+,2,↑
8	(((-	5,3,+,2,↑,8
/	(((/	5,3,+,2,↑,8
4	(((/	5,3,+,2,↑,8,4
)	(((-	5,3,+,2,↑,8,4,/
*	(((*	5,3,+,2,↑,8,4,/
((((*	5,3,+,2,↑,8,4,/
3	(((*	5,3,+,2,↑,8,4,/3
+	(((*+	5,3,+,2,↑,8,4,/3
6	(((*+	5,3,+,2,↑,8,4,/3,6
)	(((*	5,3,+,2,↑,8,4,/3,6,+
)	((5,3,+,2,↑,8,4,/3,6,+,*

Queue

1) Queue কাকে বলে? উদাহরণ সহ লিখ।

Queue হচ্ছে Linear data structure যাহা ,FIFO (First-In-First-Out) পদ্ধতিতে কাজ করে। যেই ডাটা প্রথমে Insert হবে সেই ডাটা সর্ব প্রথম Delete হবে। দুইটি FRONT এবং REAR নামক variable ধরে কাজ আমরা কাজ করে থাকি।

Application:

Waiting list : Printer, Disk, CPU

Synchronous data transfer : Pipes file IO, Sockets Handling interrupts, MP3 media player, CD player, Ticket collect ইত্যাদি।

2) Queue এর Operation লিখ।

(i) Create() (ii) Push() (iii) pop() (iv) isFull()
(v) isEmpty() (vi) peek() (vii) Clear()

3) Enqueue and Dequeue এর Algorithm লিখ।

Enqueue(Insert)	Dequeue(Delete)
1. If FRONT=1 and REAR=N Then: Print: Overflow and Return	1. If FRONT=NULL Then: Print : Underflow and Return
2. If FRONT=NULL Set FRONT:=REAR:=1	2. Set ITEM=QUEUE[FRONT]
Else if REAR=N Set REAR:=1	3. If FRONT=REAR Then: Set FRONT:=REAR=NULL
Else: Set REAR:=REAR+1	Else if FRONT=N Then: Set FRONT:=1
3. Set QUEUE[REAR]:=ITEM	Else: Set FRONT:=FRONT+1
4. Return	4. Return

4) Queue এর প্রকারভেদ লিখ এবং সংজ্ঞা লিখ।

(i) Normal Queue: যে কিউতে ডাটা এক দিক হতে ডাটা insert অন্য দিক হতে delete করা যা।

(ii) De-queue queue: যে ডাটা উভয় পাশ হতে ডাটা insert and delete করা যায়।

(iii) Priority queue: Priority অনুযায়ী ডাটা insert and delete করা যায়।

(iv) Circular queue: যে কিউতে ডাটা circular অনুযায়ী ডাটা insert and delete করা যায়।

Linked list

Basic operation of Doubly Linked list :

1. Insertion
2. Deletion
3. Insert Last, Insert
4. After, Delete last
5. Display forward
6. Display Backward

⇒ Linked List এর প্রকারভেদ লিখ এবং একটি করে চিত্র আঁক। Ans: Self study

⇒ বিভিন্ন প্রকার Searching technique গুলোর নাম এবং Complexity লিখ ।

Seaching Techniques:

(i)Linear Search:

Average and Worst Case : $O(n)$

(ii)Binary Search:

Average and Worst Case : $O(\log n)$

(iii)Interpolation Search: $O(\log(\log(n)))$

(iv)Hash Table : $O(n)$

⇒ বিভিন্ন প্রকার Sorting technique গুলোর নাম এবং Complexity লিখ ।

Sorting Techniques:

Bubble sort, Insertion sort, selection sort, merge sort, shell sort, Quick sort, Heap sort, Radix sort

Complexity:

Quick Sort=Merge Sort=Heap sort

= $O(n \log n)$ (best and average case)

Bubble, insertion sort= $O(n)$ -best case ,

$O(n^2)$ -average case

Selection sort= $O(n^2)$ best and average case

⇒ Array vs Linke list

Array	Linked list
Fixed sized	Dynamic sized
Insertion and deletion সহজ	Insertion and deletion কঠিন
Random access	No random access
Sequential access is faster করা যায়	Sequential access is slow
Binary search and linear search	Linear seach করা যায়

Stored consecutively

Stored randomly

⇒ Sparse Matrix কি

যে matrix এ Non-zero এর চেয়ে zero value এর elements বা numbers বেশি সেই matrix কে Sparse Matrix বলে

⇒ Primitive data type (PDT) কি

Predefined types of data is called PDT

Examples : integer, float character, string etc

⇒ Abstact data type (ADT) কি

Abstract data type is a mathematical model with a collection of operations defined on that model.

Examples:

Stack, Queue ,tree, graph ,linked list etc.

⇒ Overflow, Underflow, Garbage Collection, Hashing , Articulation point কাকে বলে?

Ans: Self Study

রচনায় ও সম্পাদনায়:

ENG. MD. AL FOYSAI RABBI (REHAN),

CSE,DUET,GAZIPUR

FB PAGE: (i)Fury Tent

(ii) ENGLISH CARE

Special Thanks to Ohiduzzaman