

Cursul nr. 2

The background is a dark blue gradient. It features an abstract pattern of small squares in various colors (pink, orange, teal, and light blue) and thin white vertical lines of varying lengths, scattered across the slide.

STRUCTURI DE DATE

tipuri

Lector dr. Dorin IORDACHE

Agenda



01

Liniare



02

Neliniare

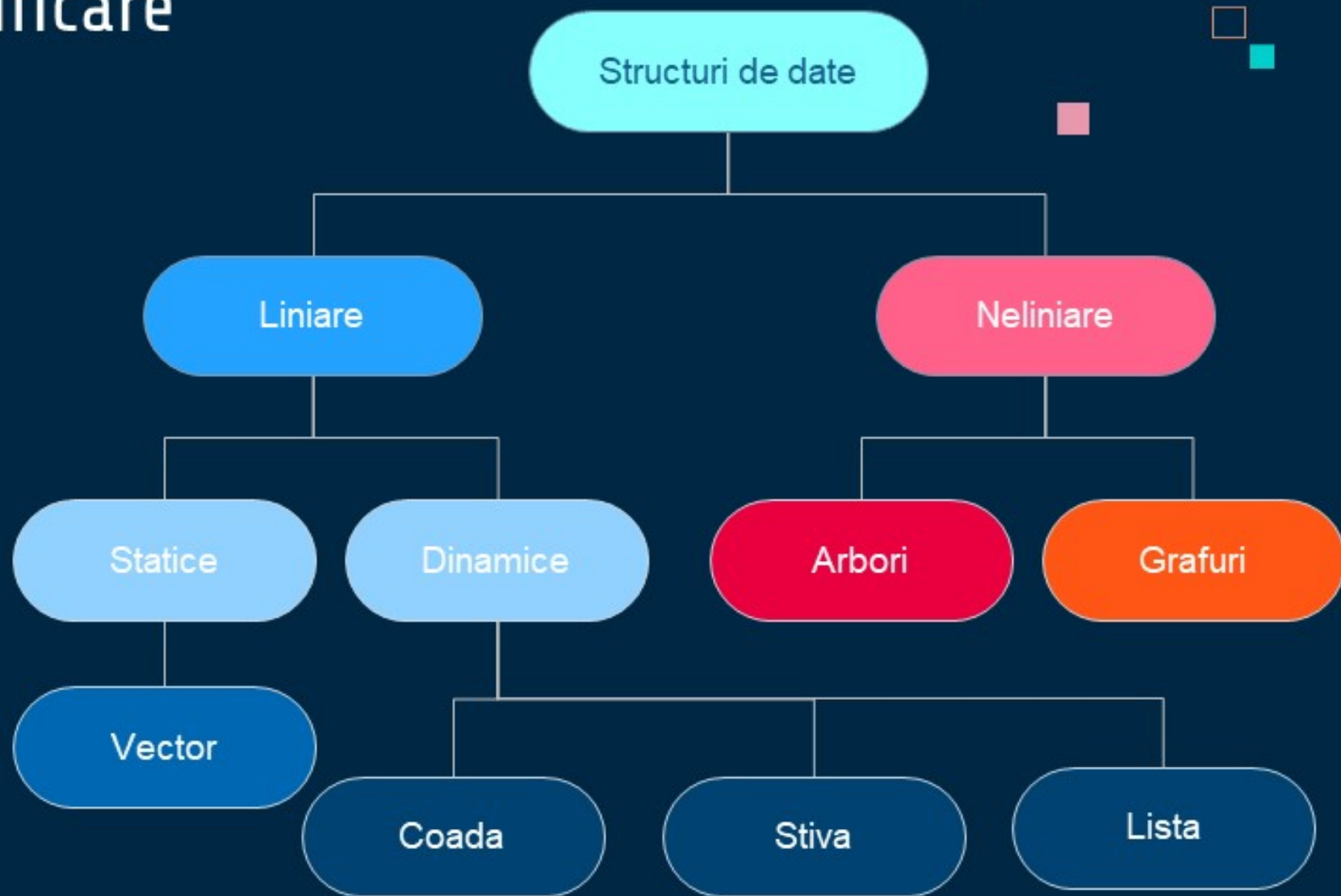
Structuri de date

= utilizate pentru stocarea și organizarea datelor

- Este o modalitate de a aranja datele astfel încât să poată fi accesate și actualizate eficient.

Structura de date și tipul de date sunt diferite.
Structura datelor este colecția de tipuri de date aranjate într-o anumită ordine.

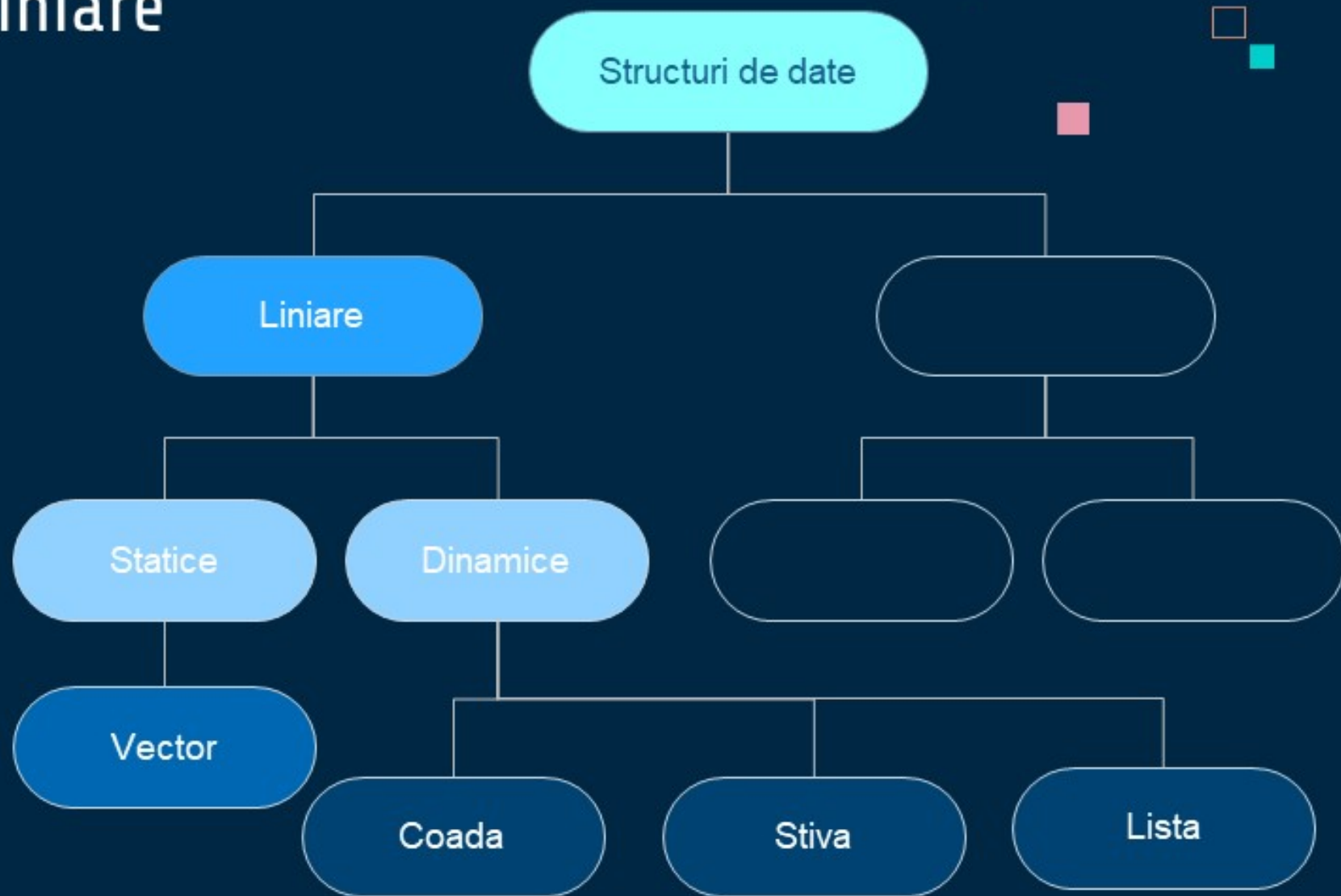
Clasificare



Structuri de date liniare

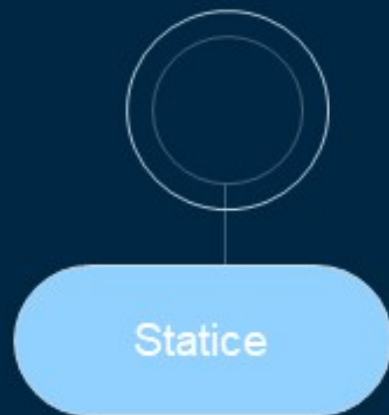
01

SD Liniare



Structuri de date liniare

elementele sunt aranjate succesiv unul după altul





Vector (Tablou)

- Unidimesnional
- Bidiminesional
- Multidimensional

Elementele vectorului din memorie sunt dispuse în zona contiguă.
Toate elementele unui tablou sunt de același tip

Vector (Tablou)

- **unidimensional**

Elemente	5	8	10	1	9
Index	0	1	2	3	4



↑
Dimensiunea vectorului

Vector (Tablou)

- **bidimensional**



The diagram illustrates a 2D array (vector) with 5 rows and 6 columns. The rows are indexed from 0 to 4, and the columns are indexed from 0 to 5. The array is labeled "Elemente" on the left and "Index 1" at the bottom. The dimensions of the vector are indicated by a horizontal line with blue dots and an arrow pointing to the text "Dimensiunile vectorului".

5	8	10	1	9	0
23	14	16	23	4	1
2	-21	2	45	6	2
10	12	-22	12	0	3
1	34	56	7	8	4
0	1	2	3	4	

Dimensiunile vectorului

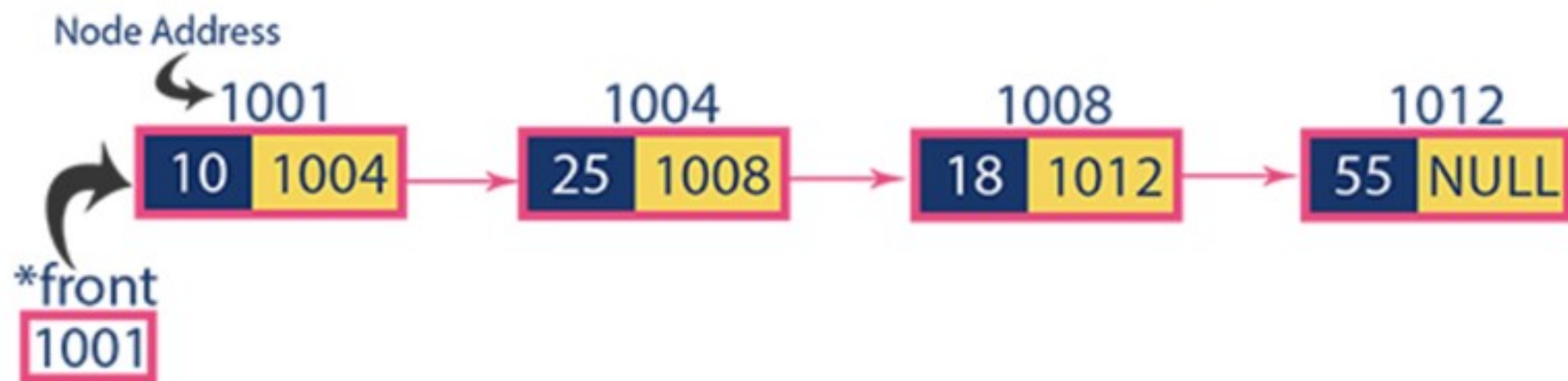
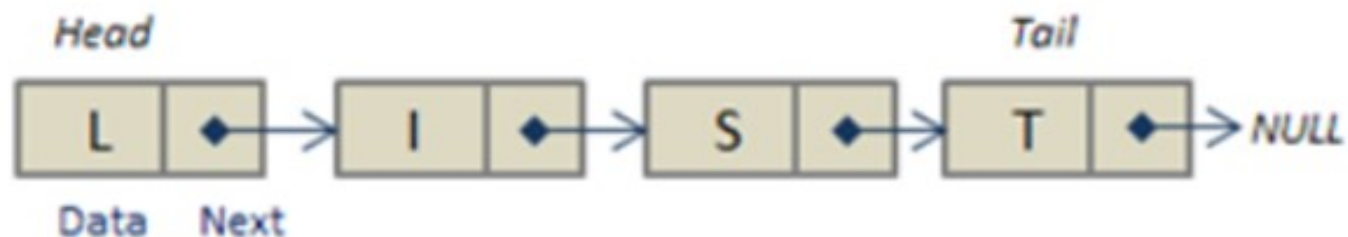


Dinamice

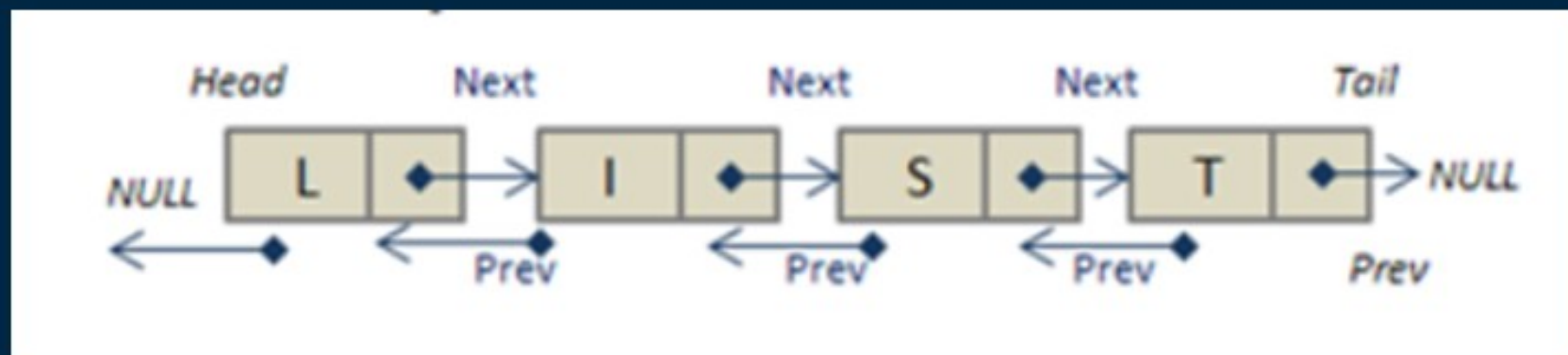
- Stiva
- Coada
- Lista

Elementele stucturilor dinamice din memorie nu sunt dispuse în zona contiguă.

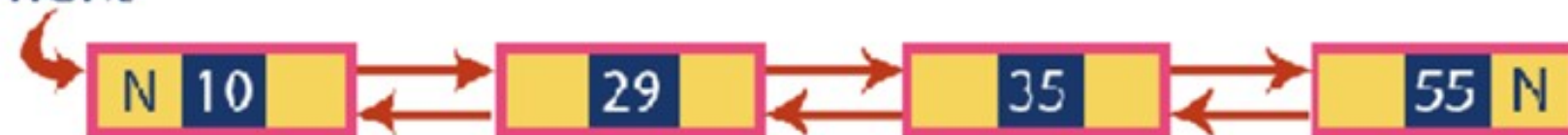
Lista simplu inlantuit



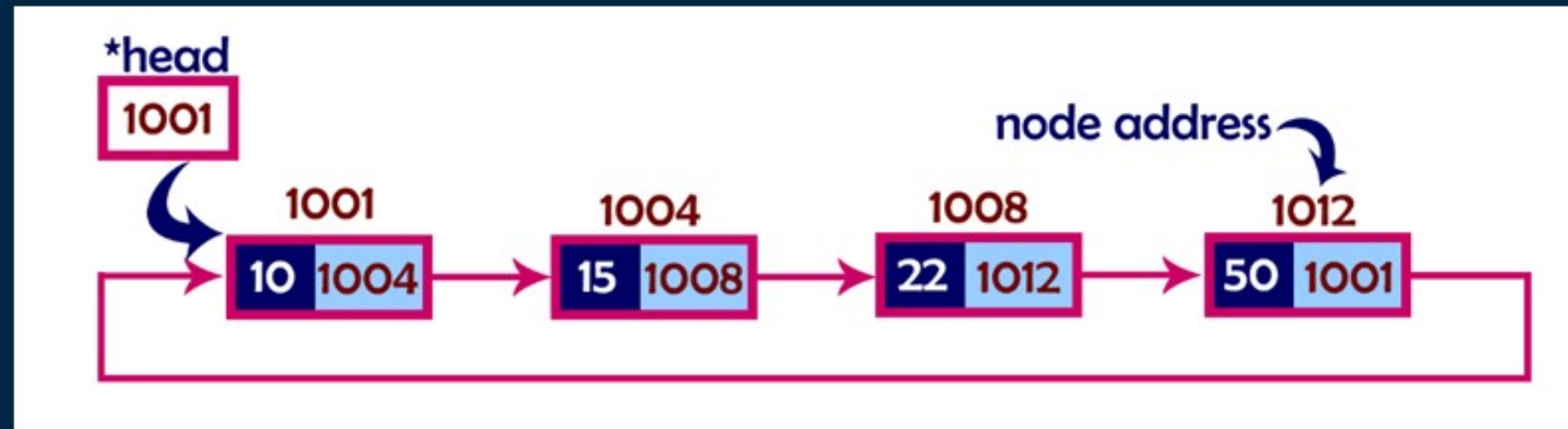
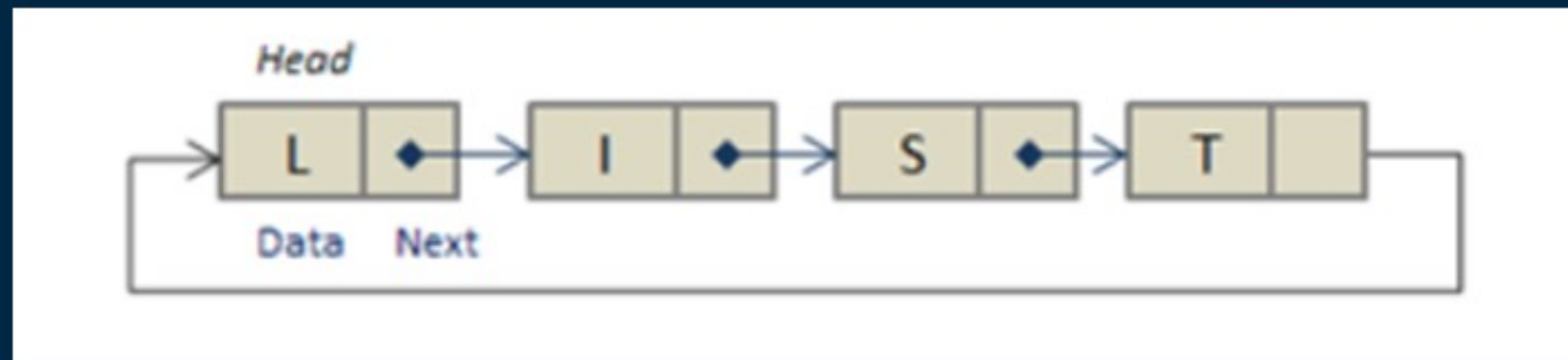
Lista dublu inlantuit

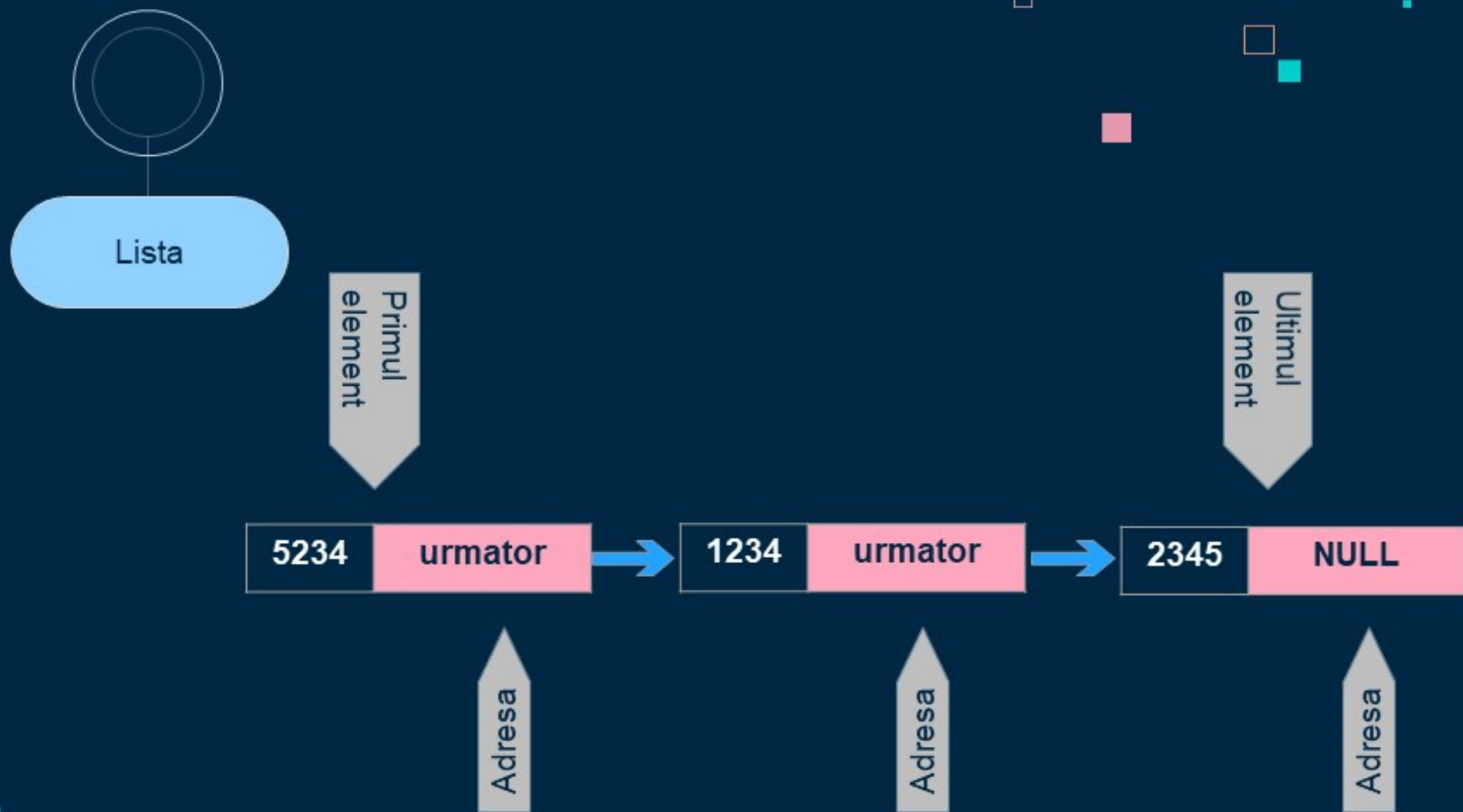


front



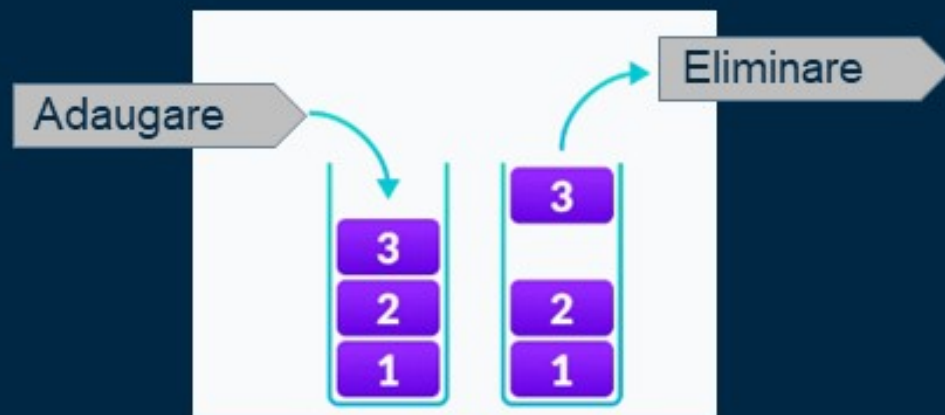
Lista







LIFO – Last In First Out





FIFO – First In First Out



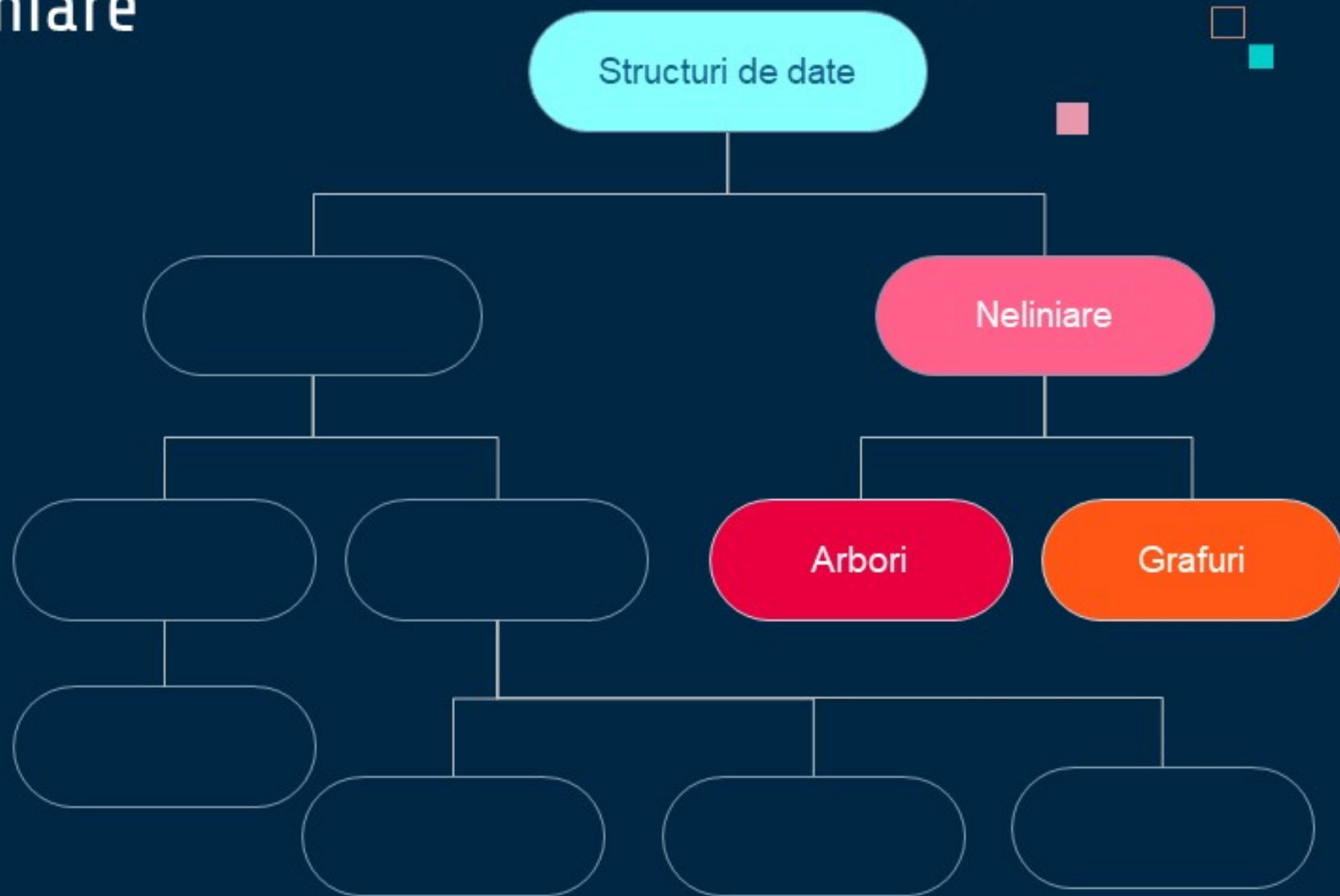
Diferente intre tablou si lista

Tablou	Lista
Tabloul este o colecție de elemente având același tip de date cu nume comun.	Lista inlantuita este o colecție ordonată de elemente care sunt conectate prin legături.
Elementele pot fi accesate aleatoriu.	Elementele nu pot fi accesate aleatoriu. Acestea pot fi accesate numai secvențial.
Elementele tabloului pot fi stocate în mod consecutiv în memorie.	Elementele listei inlantuita pot fi stocate în orice loc disponibil, deoarece adresa nodului este stocată în nodul anterior.
Operația de inserare și ștergere durează mai mult timp.	Operația de inserare și ștergere nu poate dura mai mult timp. Efectuează operarea într-un mod rapid și ușor.
Memoria este alocată în timpul compilării.	Memoria este alocată în timpul rulării.
Poate fi unidimensional, bidimensional sau multidimensional.	Poate fi o listă cu legături simplă, dublă sau circulară.
Fiecare element este independent și nu are legătură cu elementul anterior sau cu locația acestuia.	Locația sau adresa elementului este stocată în partea de legătură a elementului sau nodului anterior.
Elementele nu pot fi adăugate, șterse odată ce sunt declarate.	Nodurile din lista inlantuita pot fi adăugate și șterse din listă.
Elementele pot fi modificate cu ușurință prin identificarea valorii indexului	În lista inlantuita, modificarea nodului este un proces complex.
Pointerul nu poate fi folosit în tablou. Deci, nu necesită spațiu suplimentar în memorie pentru pointer.	Pointerii sunt utilizați în lista inlantuita. Elementele sunt menținute folosind pointeri sau link-uri. Deci, necesită spațiu de memorie suplimentar pentru pointeri.

Structuri de date neliniare

02

Neliniare

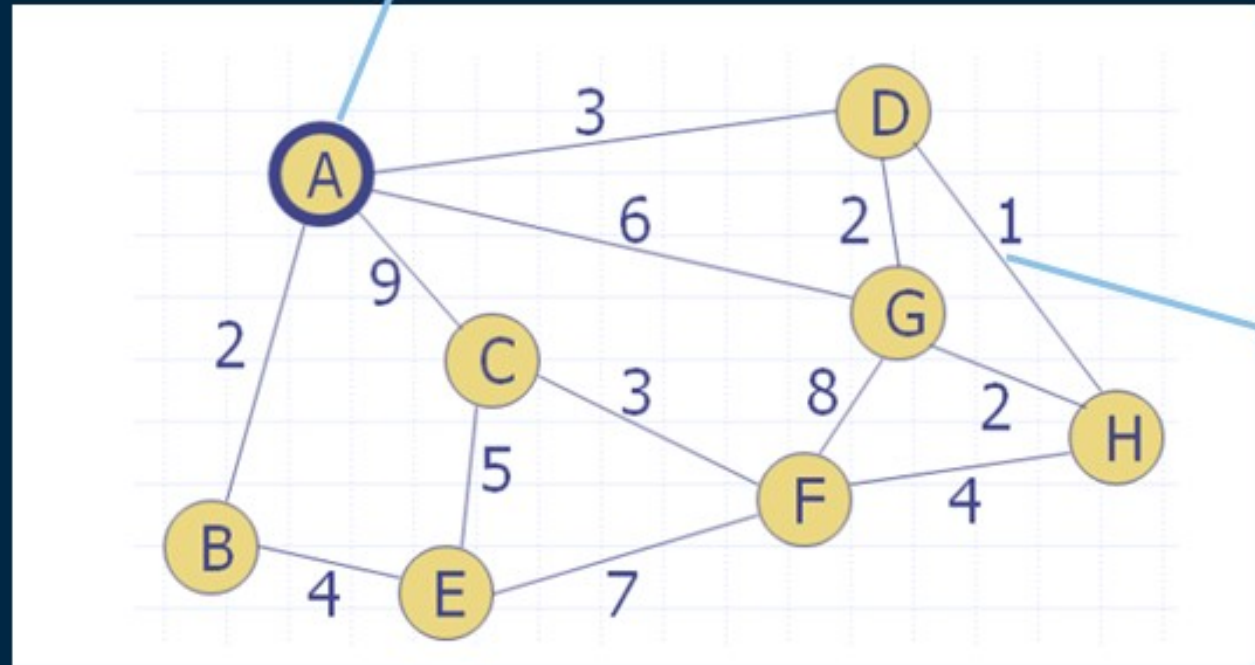




Graf

Orientat
Neorientat

Nod

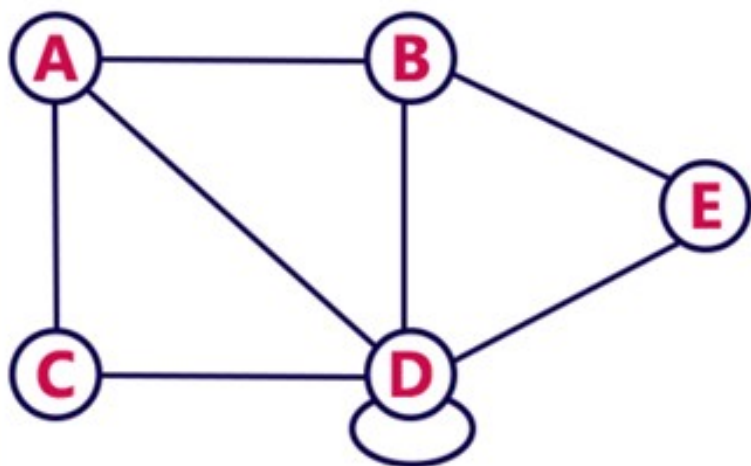


Muchie



Graf

Neorientat


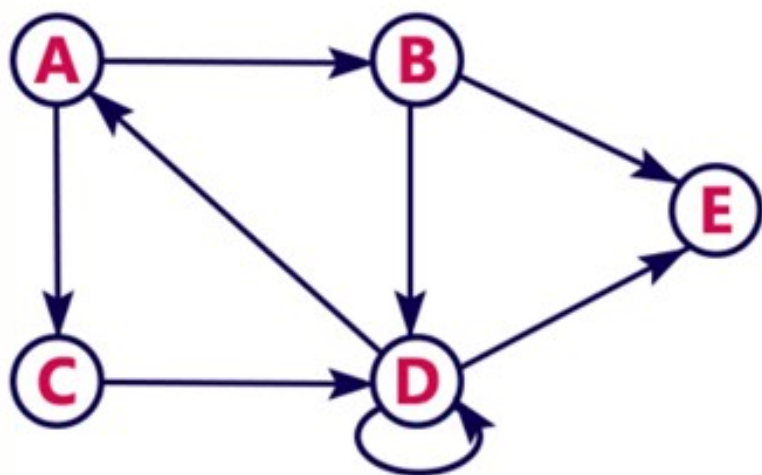


	A	B	C	D	E
A	0	1	1	1	0
B	1	0	0	1	1
C	1	0	0	1	0
D	1	1	1	1	1
E	0	1	0	1	0

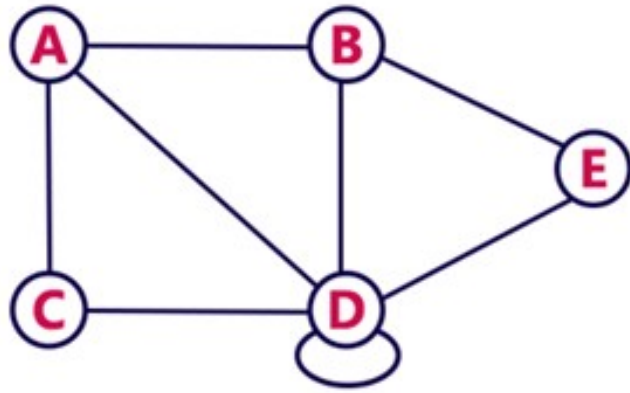


Graf

Orientat

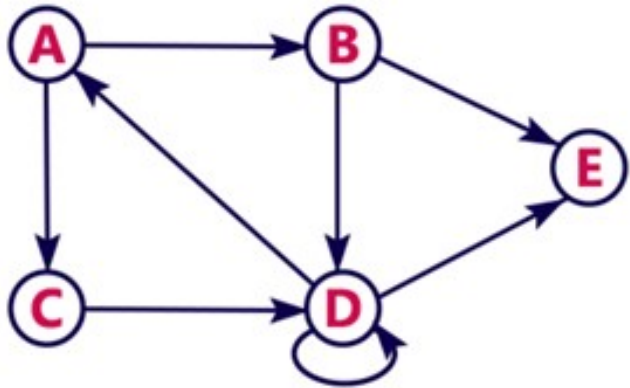


	A	B	C	D	E
A	0	1	1	0	0
B	0	0	0	1	1
C	0	0	0	1	0
D	1	0	0	1	1
E	0	0	0	0	0



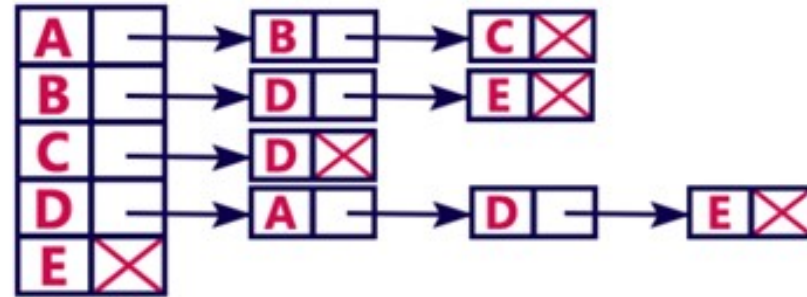
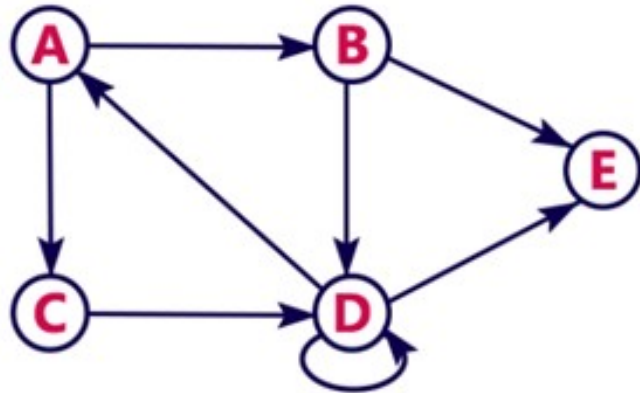
$$\begin{matrix} & \begin{matrix} A & B & C & D & E \end{matrix} \\ \begin{matrix} A \\ B \\ C \\ D \\ E \end{matrix} & \begin{pmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{pmatrix} \end{matrix}$$

Neorientat □

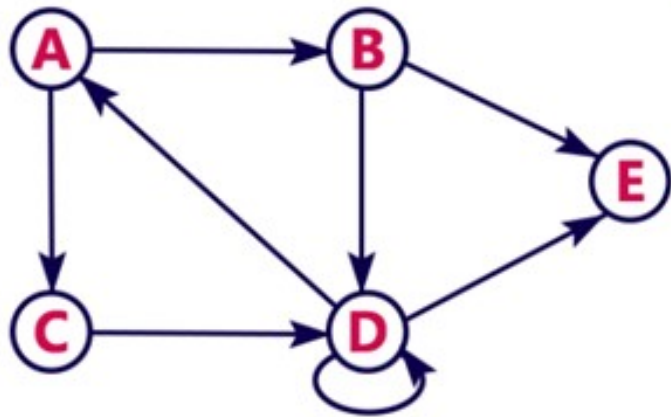


$$\begin{matrix} & \begin{matrix} A & B & C & D & E \end{matrix} \\ \begin{matrix} A \\ B \\ C \\ D \\ E \end{matrix} & \begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{matrix}$$

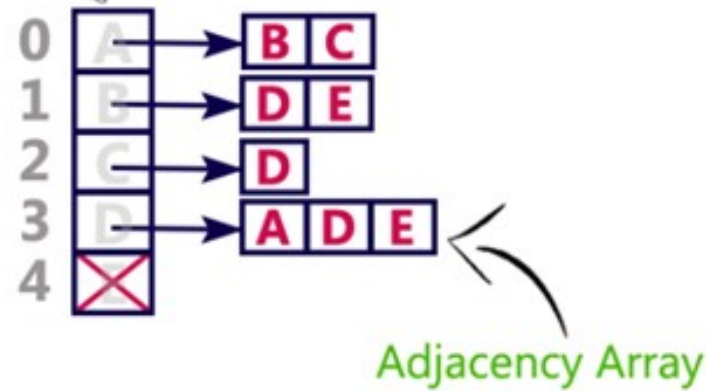
Orientat



Lista



Reference Array



Tablou



Problematici

Parcurgeri:

În lăţime : Breadth-First Search

În adâncime: Depth-First Search

Conexitate: Connected Components

Diverşi algoritmi

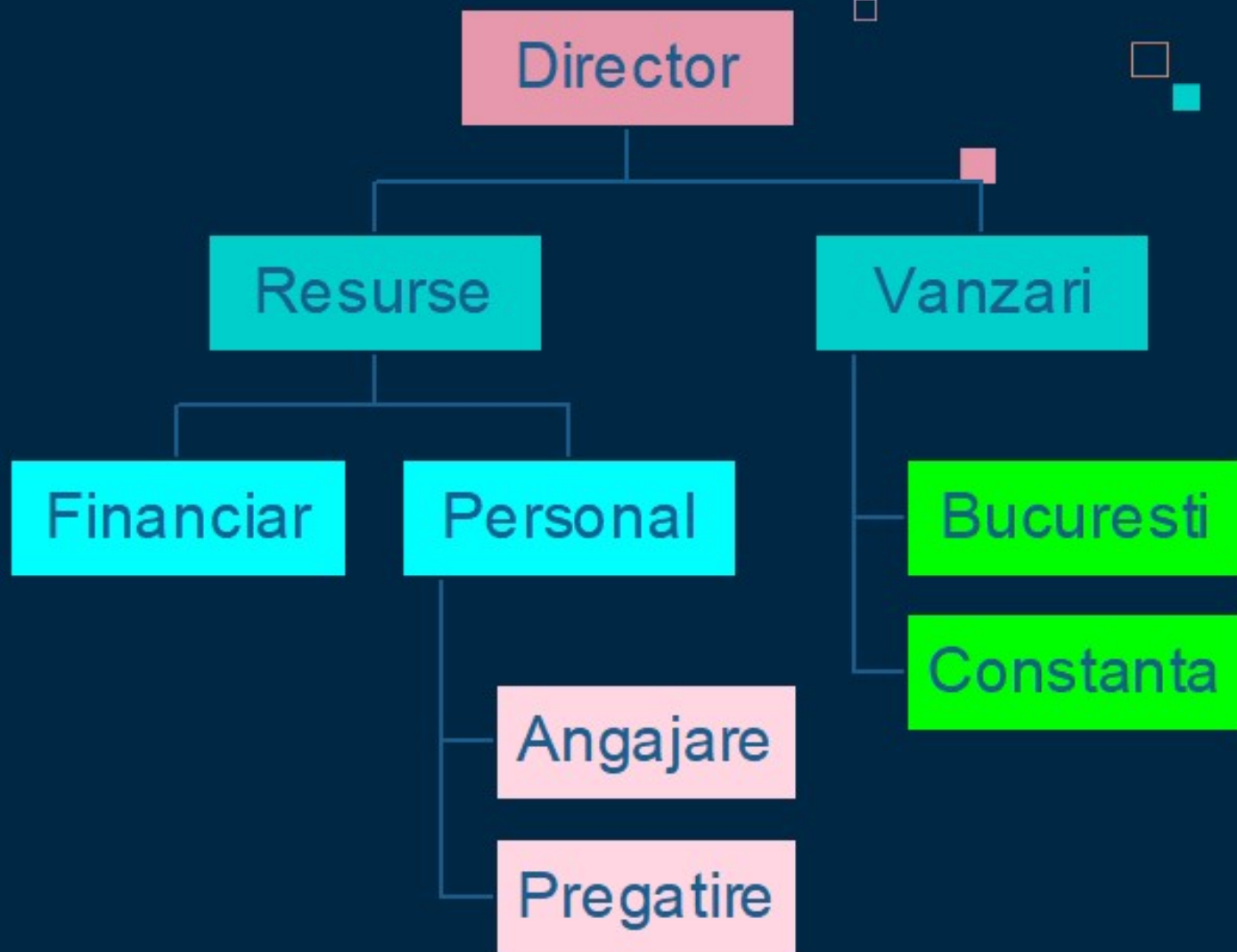
Dijkstra's Shortest Path

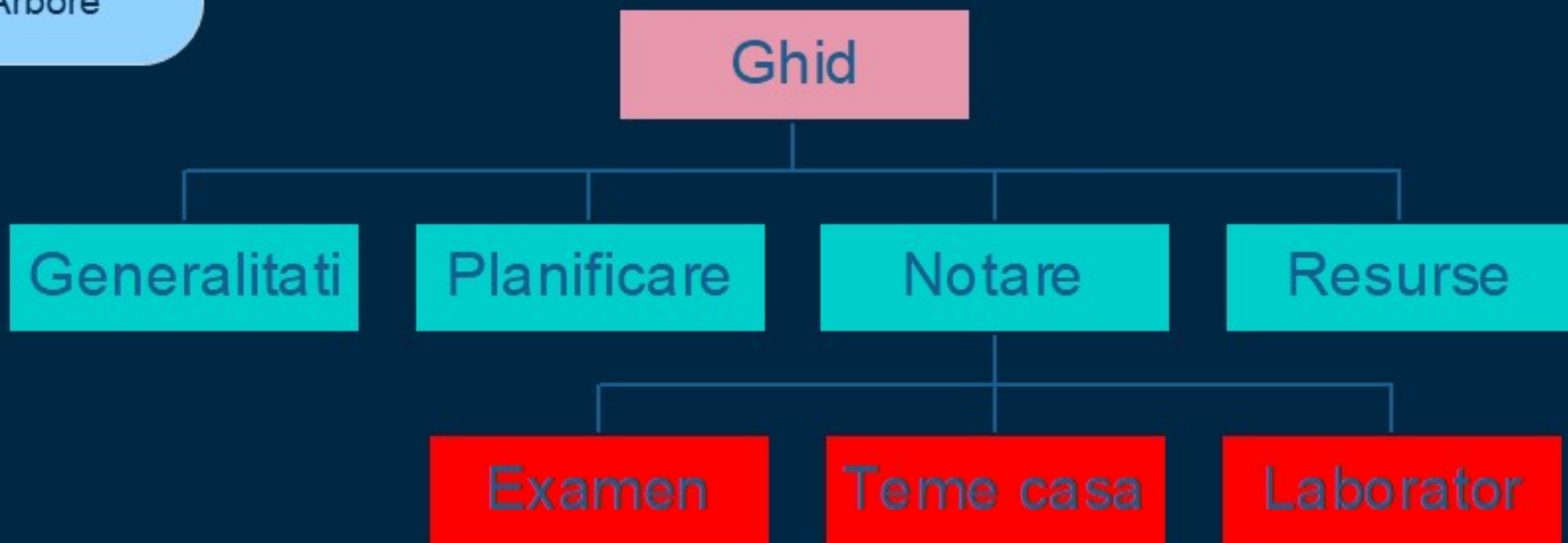
Prim's Minimum Cost Spanning Tree

Topological Sort

Floyd-Warshall (all pairs shortest paths)

Kruskal Minimum Cost Spanning Tree Algorithm



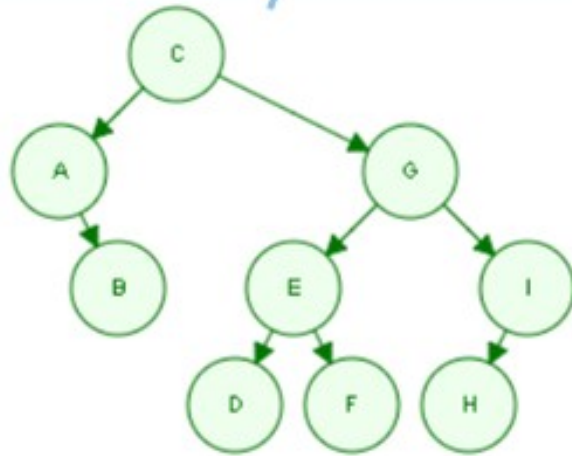


C, A, B, G, E, F, D, I, H.

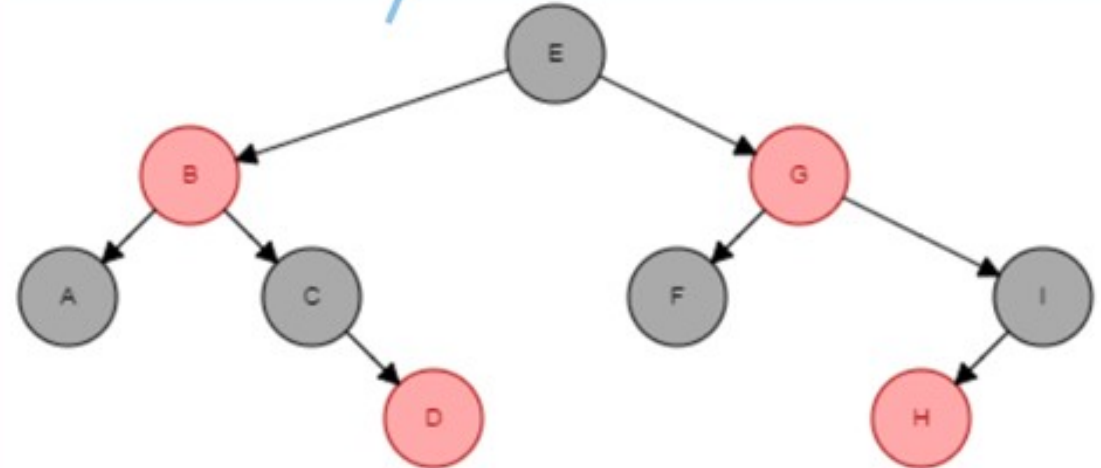


Arbore

Binar de Căutare



Roșu-Negru

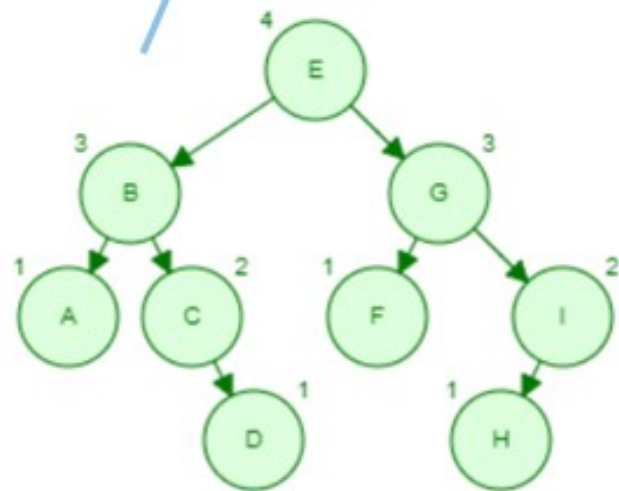


C, A, B, G, E, F, D, I, H.



Arbore

AVL



B-Tree ordin 4

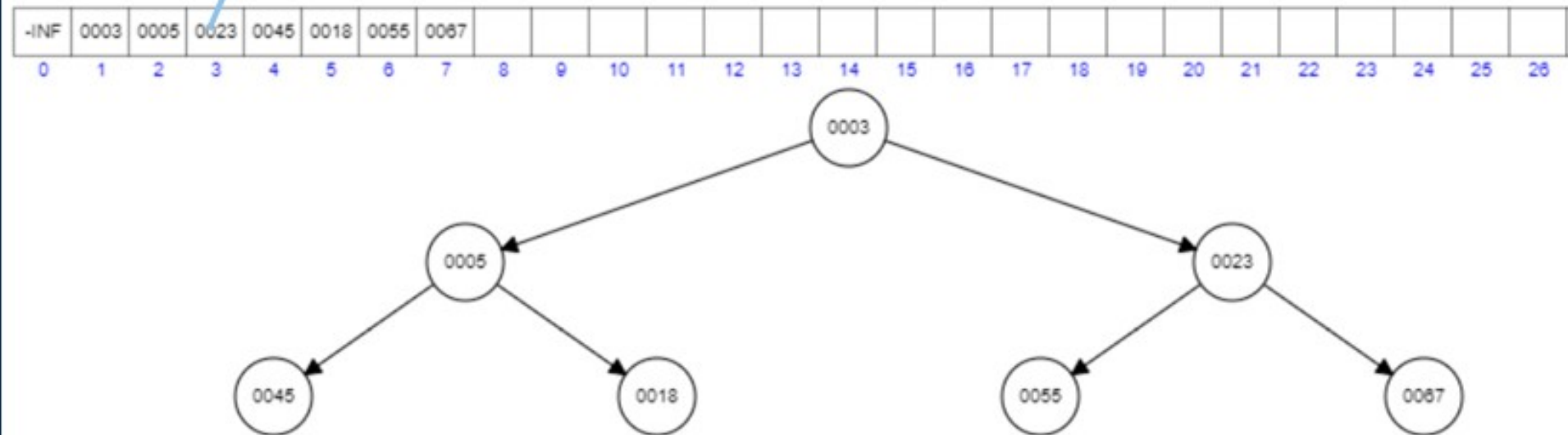


45, 23, 3, 5, 18, 55, 67



Arbore

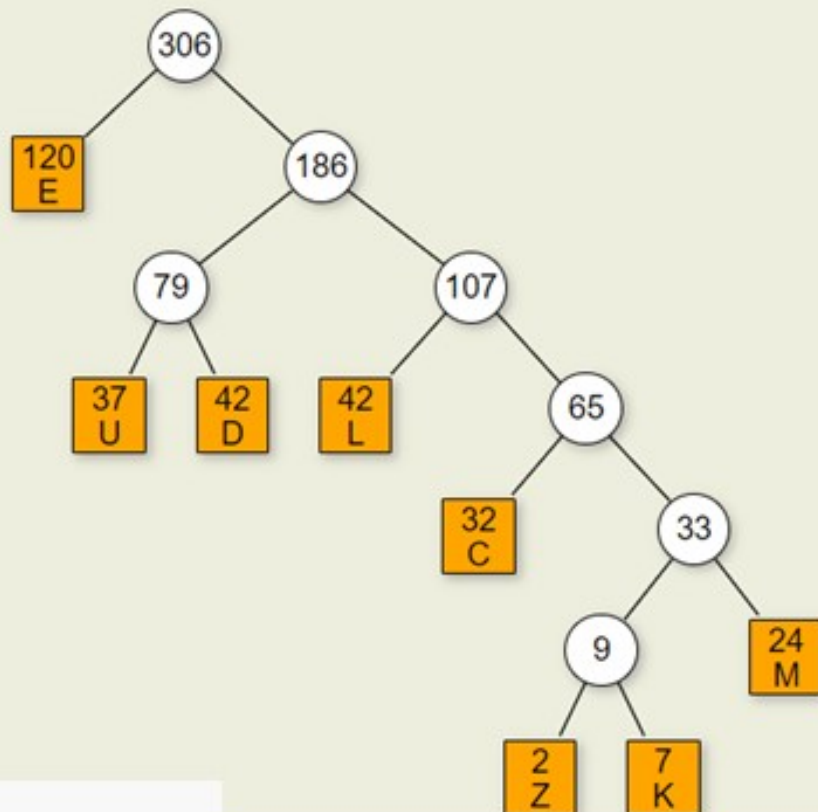
Heap





Huffman

Z	K	M	C	U	D	L	E
2	7	24	32	37	42	42	120

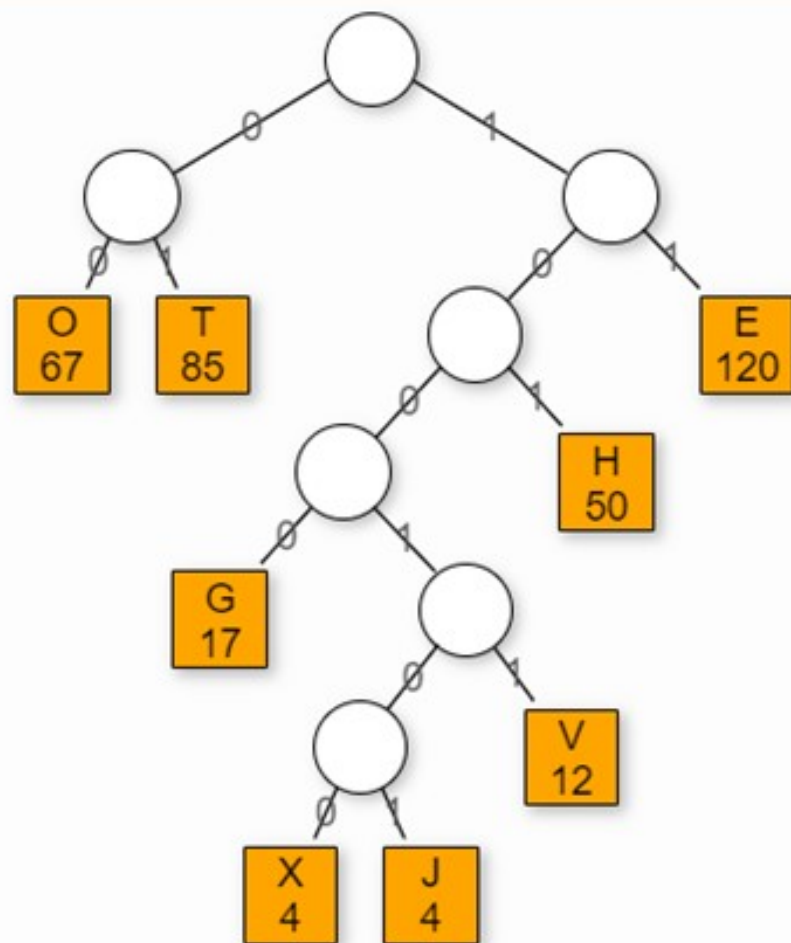


Letter	Freq	Code	Bits
E	120	0	1
D	42	101	3
L	42	110	3
U	37	100	3
C	32	1110	4
M	24	11111	5
K	7	111101	6
Z	2	111100	6

DEED 10100101 (8 bits)

MUCK 111111001110111101 (18 bits)

Huffman



THE = ?

Intrebari?

dorin.lordache@365.univ-ovidius.ro

Multumesc

CREDITS: This presentation template was created by [Slidesgo](#), including icons by [Flaticon](#), and infographics & images by [Freepik](#)