Structuri de date și algoritmi

Universitatea Politehnica Timișoara
P-ța Victoriei nr. 2
RO 300006 - Timișoara
Tel: +4 0256 403000
Fax: +4 0256 403021
rector@rectorat.upt.ro
www.upt.ro

Domeniul de studii: Informatică/ Specializarea: Informatică

SDA – Cursul 8

Ș.l. dr.ing. Adriana ALBU
adriana.albu@upt.ro
http://www.aut.upt.ro/~adrianaa

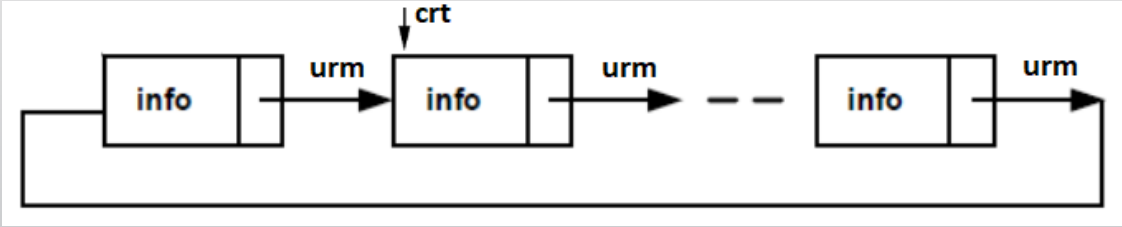
Universitatea
Politehnica
Timişoara

P-ţa Victoriei nr. 2
RO 300006 - Timişoara
Tel: +4 0256 403000
Fax: +4 0256 403021
rector@rectorat.upt.ro
www.upt.ro

5. Structura de date listă
(partea a doua)

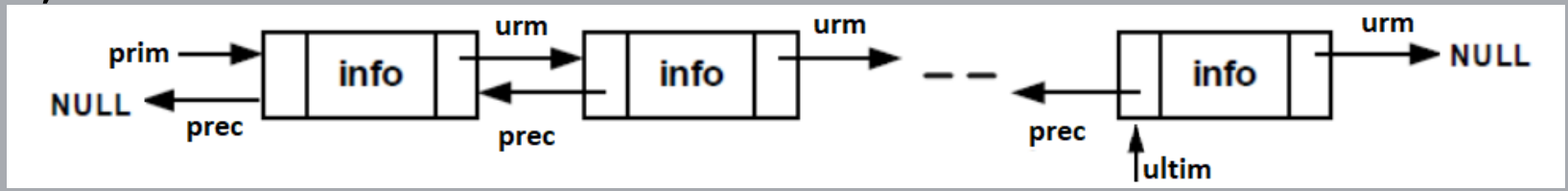
5.4 Alte tipuri de liste

1. Liste simplu înlănțuite ordonate
2. Liste circulare



3. Liste dublu înlănțuite

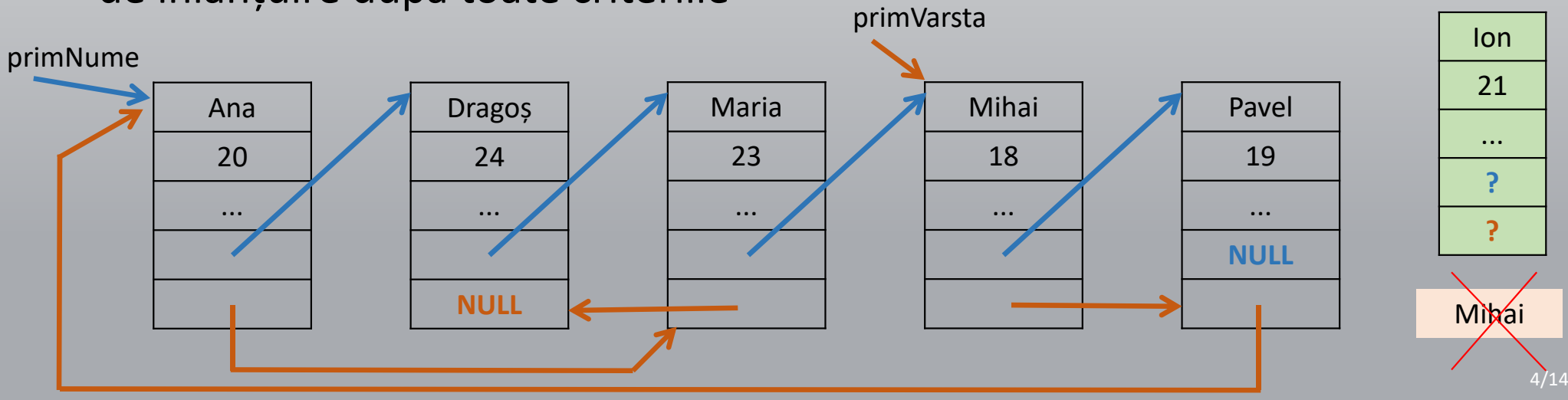
```
typedef … TipElement;
typedef struct nod{
    TipElement elem;
    struct nod *prec;
    struct nod *urm;
}TipNod;
```



5.4 Alte tipuri de liste

4. Liste multiplu înlănțuite

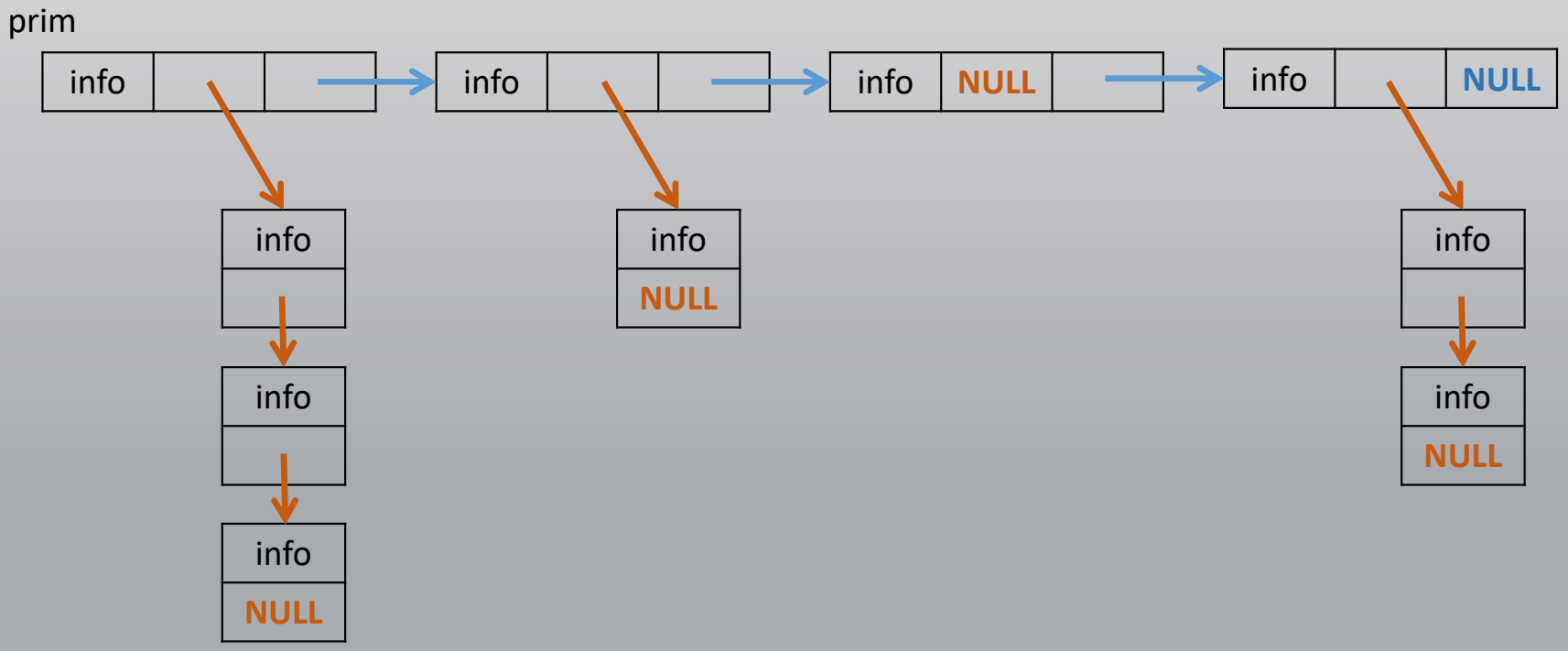
- structura nodului conține **mai multe câmpuri de înlănțuire**, asociate unor criterii diferite de ordonare
- acestui tip de lista i se asociază un număr de **pointeri de început** egal cu numărul câmpurilor de înlănțuire
- toate operațiile de **inserare și ștergere** trebuie să refacă valoarea câmpurilor de înlănțuire după toate criteriile



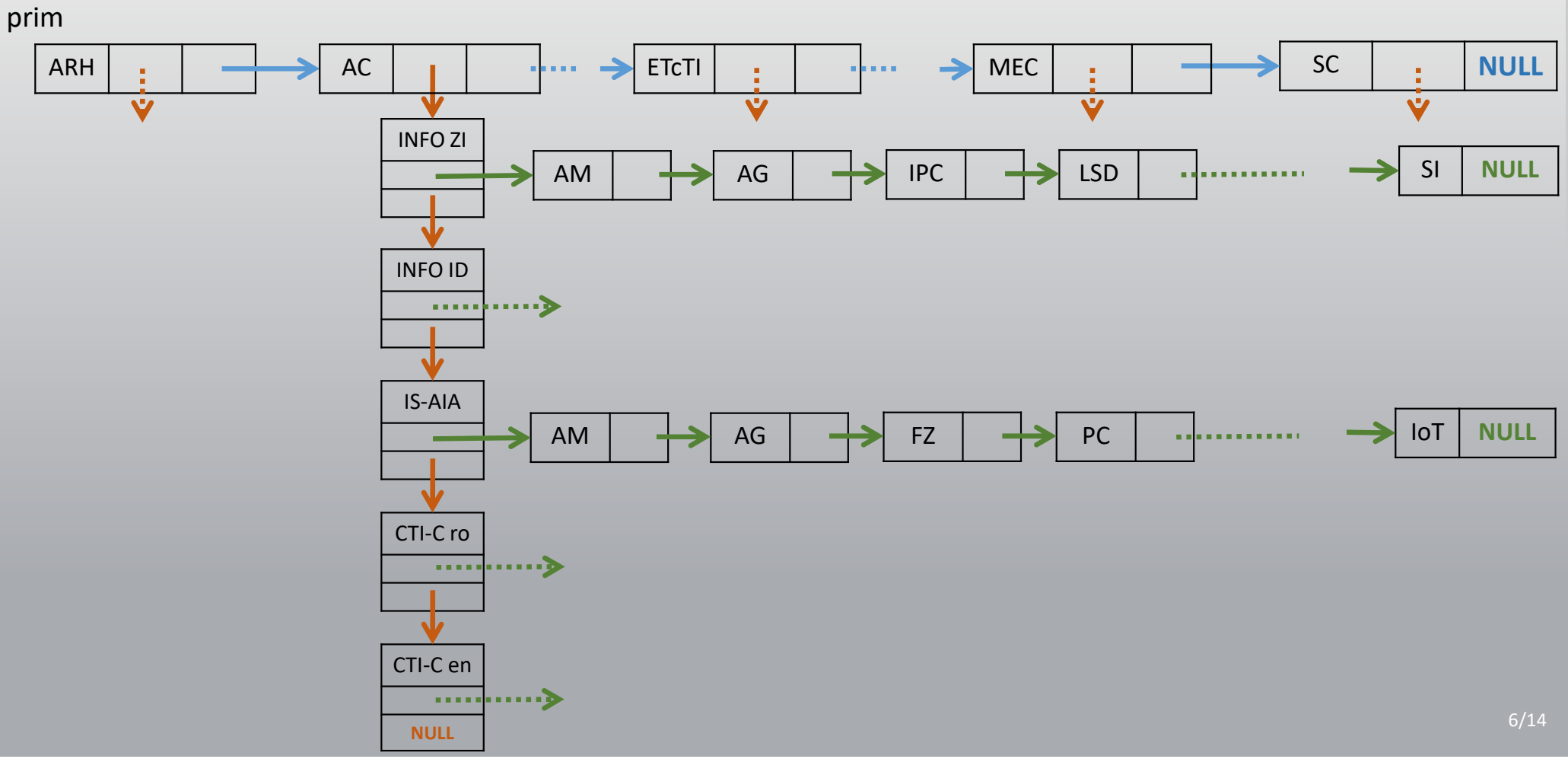
5.4 Alte tipuri de liste

5. Liste generalizate. Liste cu sub-liste

- uzual se identifică o listă principală și sub-liste asociate nodurilor acesteia
- uzual tipul nodurilor din lista principală diferă de tipul nodurilor din sub-liste



Liste cu sub-liste – Exemplu



5.5 Structuri derivate din tipul listă / 5.5.1 Structura de date stivă

- TDA stivă
- MM
  - tip special de listă
  - inserările și ștergerile se fac într-un singur capăt, numit uzual vârful stivei
  - se aplica principiul „ultimul intrat – primul ieșit” (LIFO – *last in first out*)
- Notații
  - TipStiva s;
  - TipNod x;
  - boolean b;
- Operatori
  - Initializare(s);
  - TipNod VarfStiva(s);
  - TipNod pop(s);
  - push(s,x);
  - boolean StivaVida(s);
- Implementare: tablouri; pointeri

5.5.2 Structura de date coadă

- TDA coadă
- MM
  - tip special de listă
  - elementele sunt inserate la un capăt al cozii (după nodul „ultim”) și sunt suprimate la celălalt capăt al cozii (nodul „prim”) – FIFO – *first in first out*
- Notații
  - TipCoada c;
  - TipElement x;
  - boolean b;
- Operatori
  - Initializare (TipCoada c);
  - TipElement Prim(TipCoada c);
  - adauga(TipCoada c, TipElement x);
  - extrage(TipCoada c);
  - boolean CoadaVida(TipCoada c);
  - boolean CoadaPlina(TipCoada c);
    - pentru implementarea cu tablouri

Implementare coada

a) Pointeri – doi pointeri (prim, ultim)

```
typedef … TipElement;
typedef struct nod{
    TipElement element;
    struct nod *urm;
}TipNod;
typedef tipNod* TipPointerNod;
typedef TipPointerNod{
    TipPointerNod prim, ultim;
}TipCoada;
```

Implementare coada

b) Tablouri

- Liniare
  - adăugarea se face în O(1), iar ștergerea în O(n)

- Circulare
  - adăugarea și ștergerea se realizează în O(1)
  - inserarea presupune avansul indicatorului „ultim”, iar ștergerea avansul indicatorului „prim”
  - coada se rotește în sensul acelor de ceasornic după cum se adaugă sau se scot elemente din ea

Cozi bazate pe prioritate

- Tip special de coadă:
  - elementelor din structură li se asociază o prioritate, iar la extragere va fi extras (șters) de fiecare dată elementul cu prioritatea cea mai mare
- Operatori:
  - inserare;
  - extragere (elementul cu prioritatea cea mai mare);
  - schimba_prioritate;
  - suprima (șterge) un element oarecare;
  - reuneste doua cozi;
  - inlocuire
    - se înlocuiește „cel mai mare” element cu un nou element
    - constă de fapt dintr-o inserare urmată de extragerea „celui mai mare”
  - schimba
    - o suprimare urmată de inserare

Implementare cozi cu prioritate

a) Liste neordonate (cu pointeri, cu tablouri)

```
void insereaza(Tip Elem x){
    N++; a[N-1]=x;     // O(1)
}
TipElem extrage(){
TipElem x;
int j, max=0;
    for(j=1;j<N;j++)
        if(a[j].cheie>a[max].cheie) max=j;//O(N)
    x=a[max];
    a[max]=a[N-1]; N--;
    return x;
}
```

Implementare cozi cu prioritate

b) Liste ordonate (cu pointeri, cu tablouri)
- inserare -> O(N)
- extragere -> O(1)

c) Ansamblu – arbore binar, parțial ordonat, implementat cu tablou
- $a_i >= a_{2i}$
- $a_i >= a_{2i+1}$
- i=1…N/2

➢ Obs. Operațiile implementate pe structura de tip coadă cu prioritate pot fi utilizate pentru implementarea unor algoritmi de sortare
- spre exemplu utilizarea repetată a operației de inserare urmată de extragere conduce la obținerea unei secvențe sortate

Universitatea
Politehnica
Timişoara

P-ţa Victoriei nr. 2
RO 300006 - Timişoara
Tel: +4 0256 403000
Fax: +4 0256 403021
rector@rectorat.upt.ro
www.upt.ro

Vă mulțumesc!