

sda-f@lists.deae.pub.ro

lists.deae.pub.ro

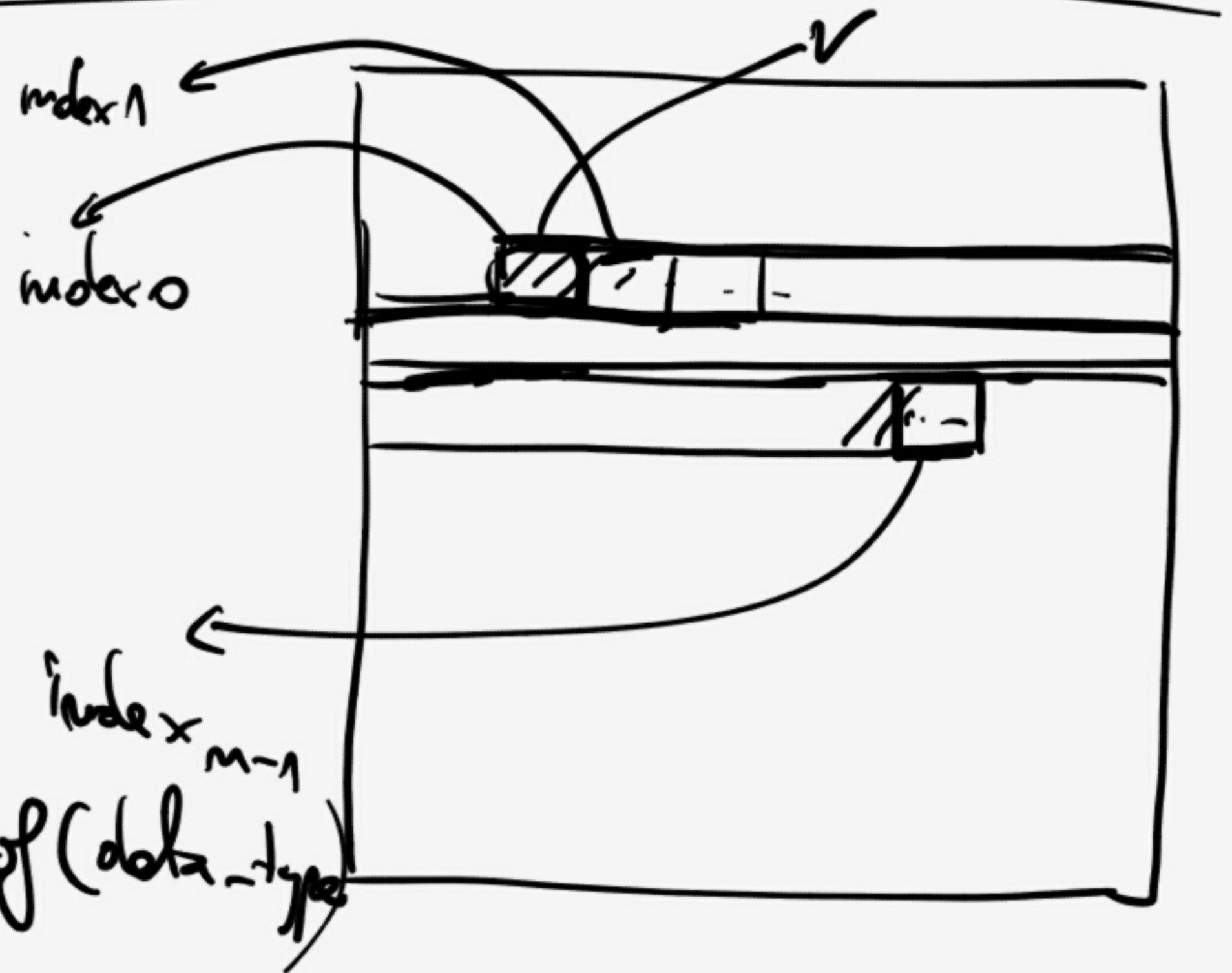
wiki.deae.pub.ro

Liste de date

Vector

- dimensiune fixă

$$\text{dimensiunea}(V) = n * \text{sizeof}(\text{data_type})$$

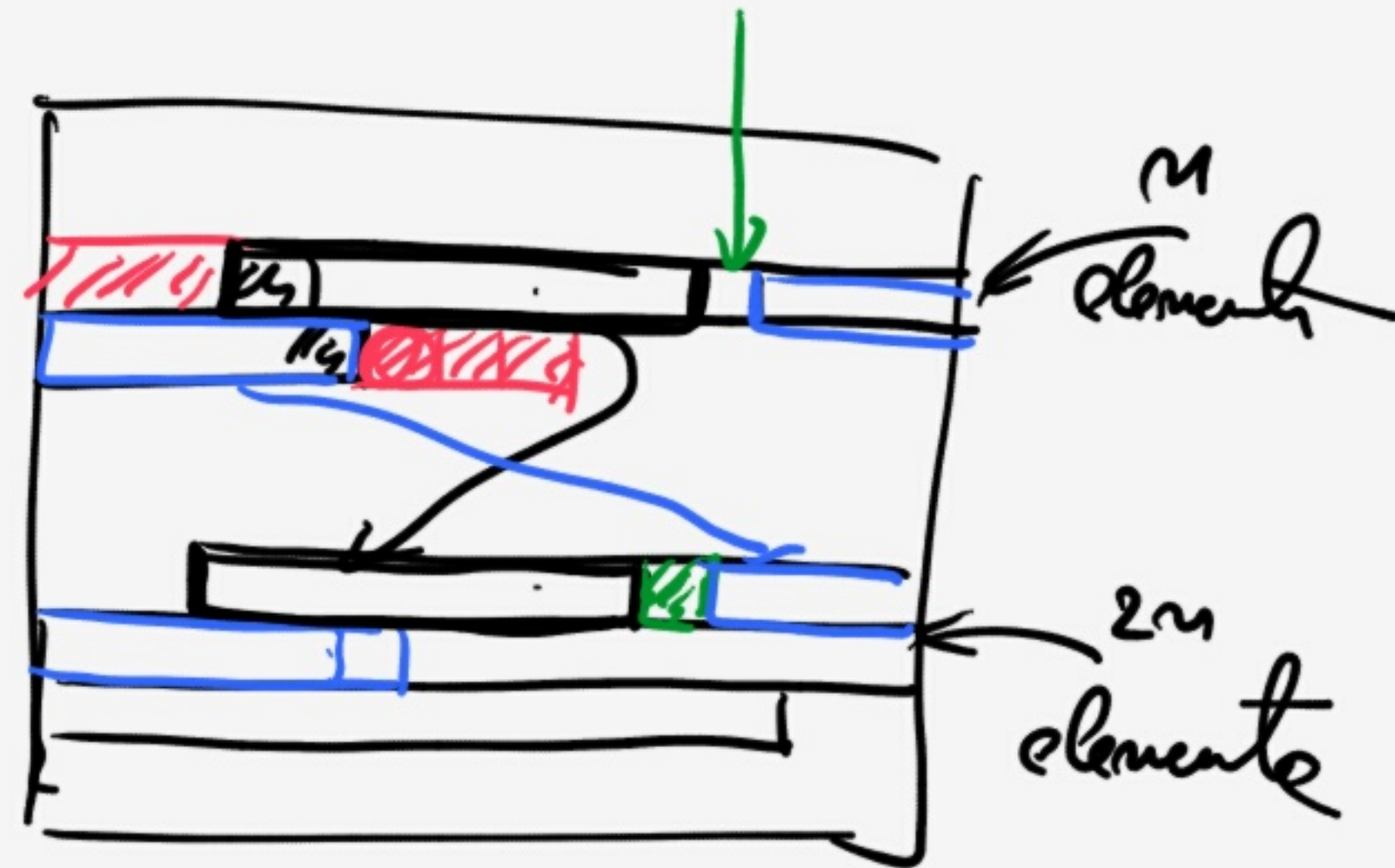


① Inserție în vector $O(n)$

a) Realocare (2x) → memcpy

b) Copiere

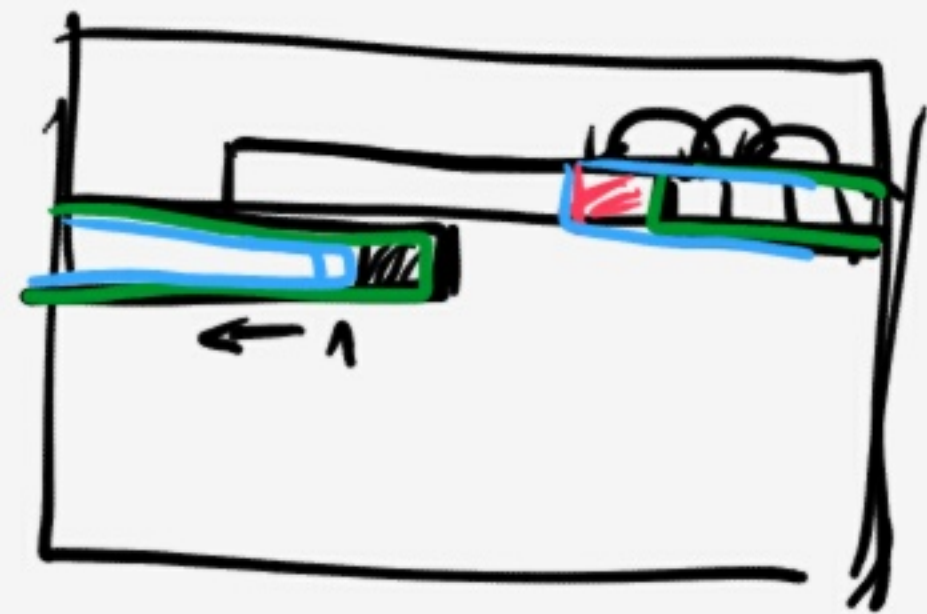
`memcpy(dest, sursa, nize);`



`memcpy(newV, oldV, k * nizeof(dob+));`

② Ștergere din vector $O(n)$

a) Copiere → memmove



③ Căutare

a) Secvențială \rightarrow for $i=0 \dots n-1 \rightarrow O(n)$

b) Binară $\rightarrow O(\log_2 n)$

| | | | | | | | | | | | |
|---|---|----|----|----|----|----|----|----|-----|-----|-----|
| 2 | 7 | 20 | 23 | 23 | 25 | 30 | 50 | 51 | 100 | 103 | 120 |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |

Note: In the original image, the value 20 at index 2 is highlighted with a thick border. A bracket under indices 5-11 is crossed out with a diagonal line. An arrow points from the text $O(\log_2 n)$ to the value 20.

$\rightarrow 20?$

0-11 \rightarrow 25 \neq 20

0-4 \rightarrow 20 \neq 20 \rightarrow 2

④ Sortarea
- bubble-sort $\rightarrow O(n^2)$
= ?
= .

Stiva

indexul de citire indexul de scriere

① Cu vector:

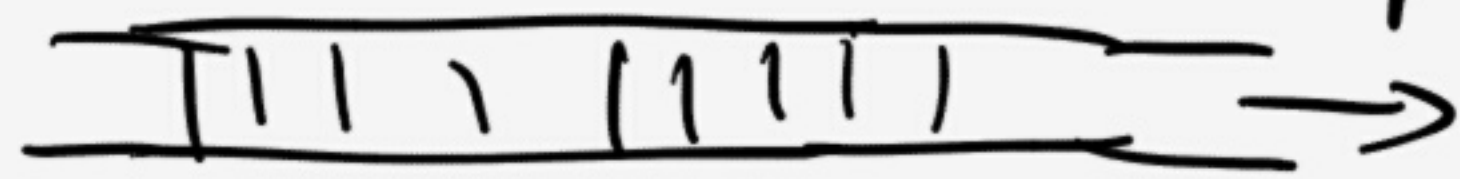
LIFO

a) push $\rightarrow O(1)$
 $\rightarrow V[w_i] = \text{new_value}$

b) pop \rightarrow value = $V[r_i]$; $\rightarrow O(1)$
 $w_i++;$
 $r_i++;$
 $w_i--;$

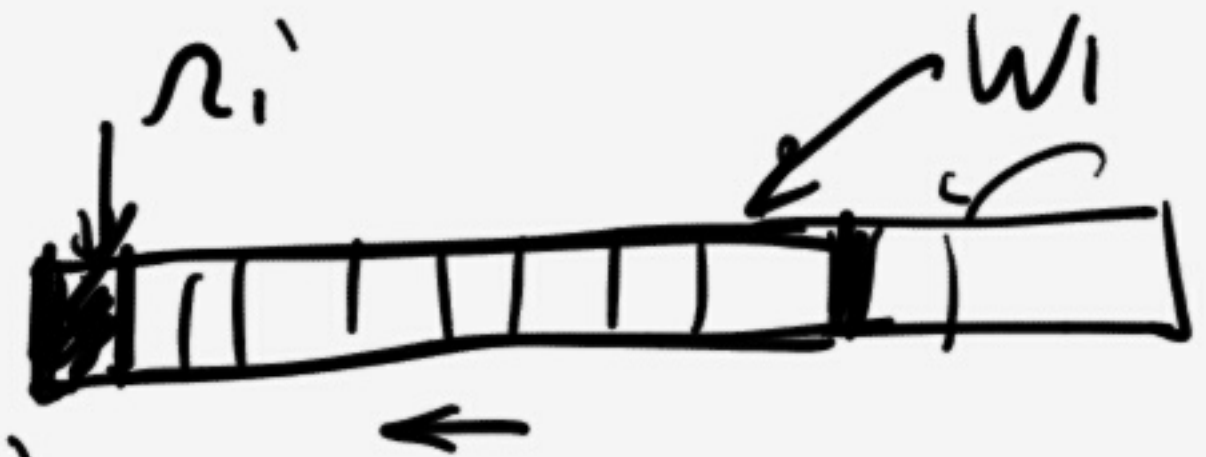
Coada - FIFO

push

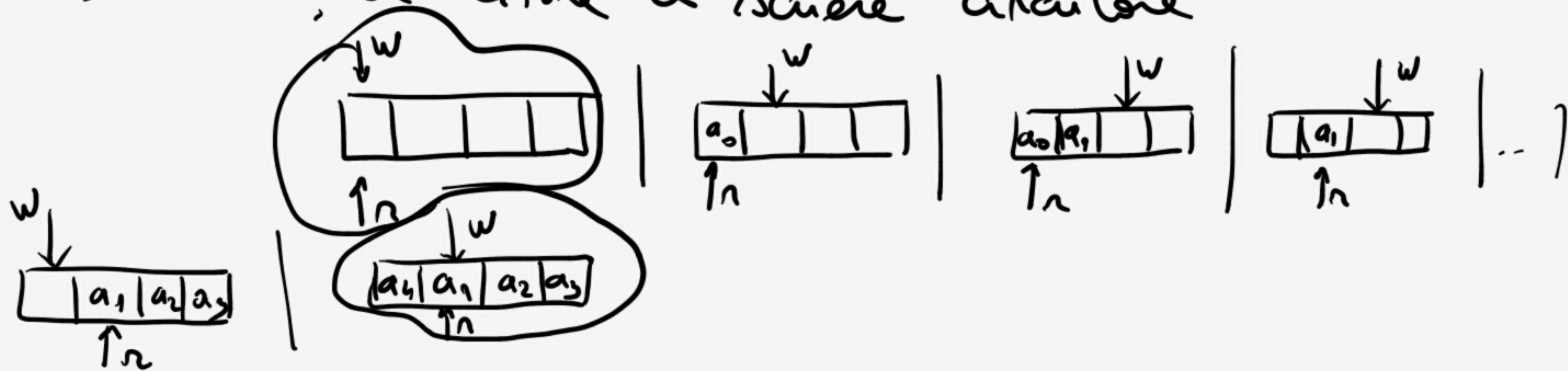


pop

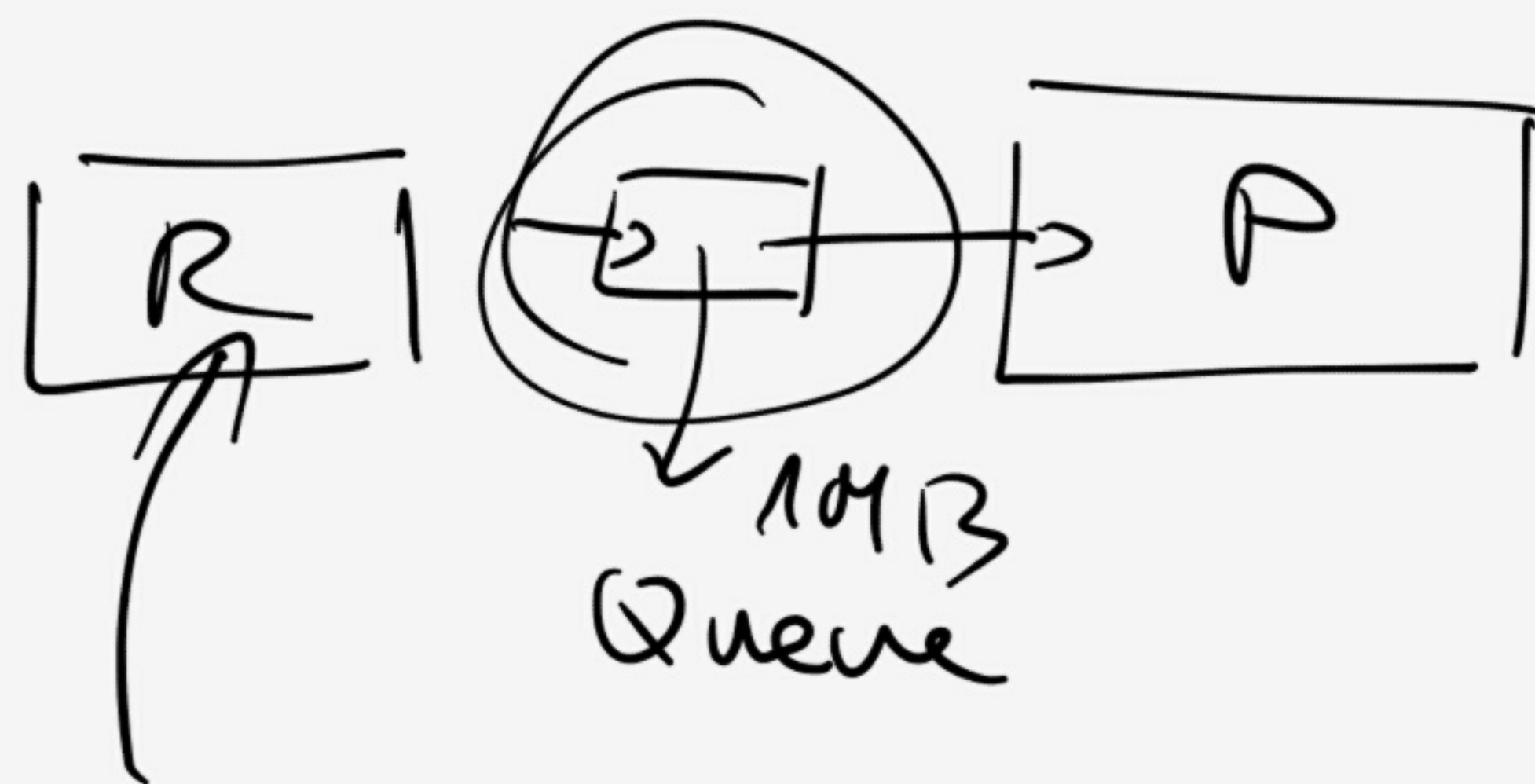
- a) → investire pentru push $O(1)$
→ eliminare pentru pop $O(n)$



- b) indicii de citire & scriere circulare



()



Struct → tip de dată ≃ int

```
int a;  
int *a;
```

struct caine

struct caine fiob;

struct caine *plex;

operato pentru
acces la
membrii unei
structuri

```
struct caine {  
    char * rasa;  
    char * nume;  
    unsigned char varsta;  
    char * culoare;  
};
```

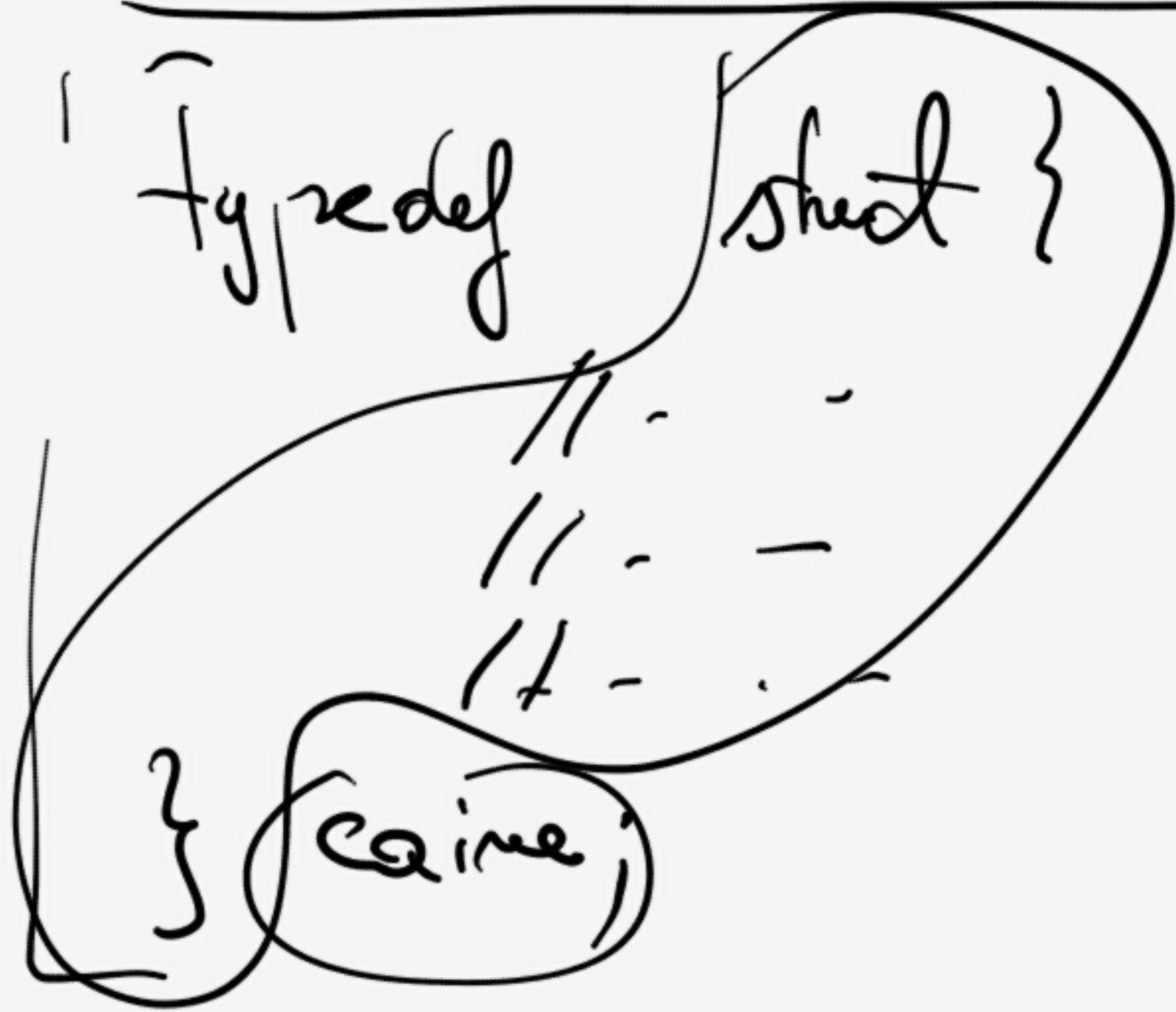
fiob.rasa = ~~"pedneas"~~;

fiob.varsta = 3;

plex = (struct caine *) malloc(
 sizeof(struct caine));

struct caime *plex = ...

(*plex).varsta = 5; \equiv plex \rightarrow varsta = 5;



\rightarrow caime fiola;


```
struct caime *pLex = (struct caime *) malloc (sizeof(struct caime));
```

```
pLex->name = (char *) malloc (4 * sizeof(char));
```

```
strcpy (pLex->name, "Lex");
```

```
//...
```

```
free (pLex->name);
```

```
free (pLex);
```