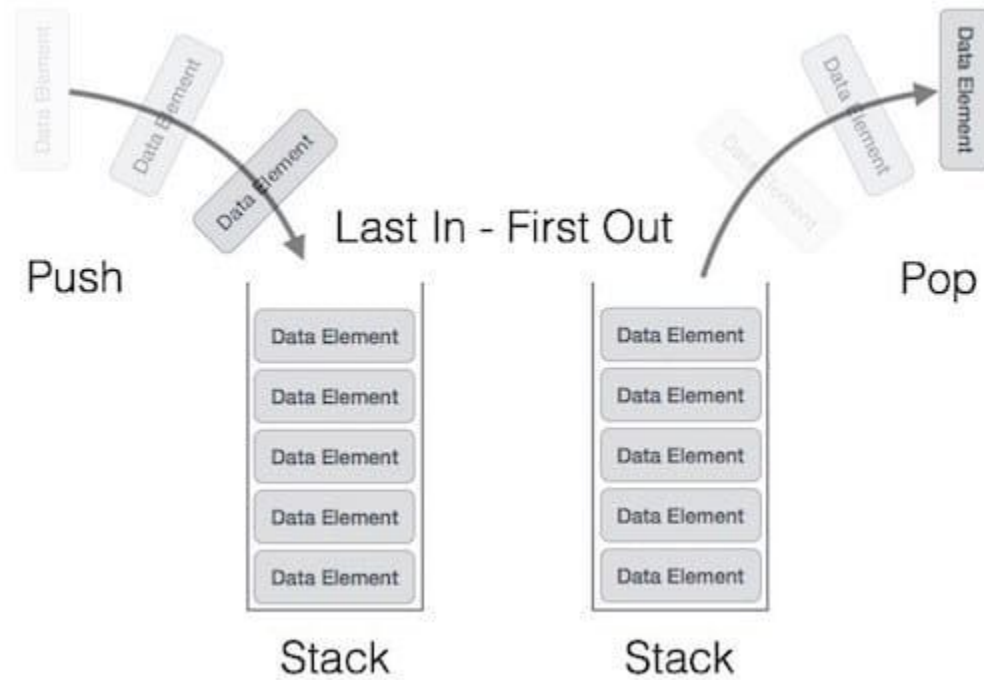

Structuri de Date Elementare

— Stive Cozi Deque Mars? Batog? —

Stive (Stack)

- Sunt structuri de date de tip **LIFO** (**L**ast **I**n **F**irst **O**ut)
- Avem acces numai la elementul din vârf (top)
- Operații de bază:
 - Push - adăugarea unui element (în vârf)
 - Pop - eliminarea elementului din vârf
- Operații suplimentare:
 - Size() - numărul de elemente
 - isEmpty() - returnează **true** dacă numărul de elemente este exact 0
 - Peek() - ne spune valoarea din vârf fără să o extragă

Stive (Stack)



Stive (Stack)

- Metode de implementare:
 - Stivă ca Vector
 - [Vizualizare](#)
 - [Implementare](#) (găsiți în secțiunea de implementare ca array)
 - Stivă ca Listă
 - [Vizualizare](#)
 - [Implementare](#) (găsiți în secțiunea de implementare ca linked list)
 - Stivă în C++ - <https://en.cppreference.com/w/cpp/container/stack>

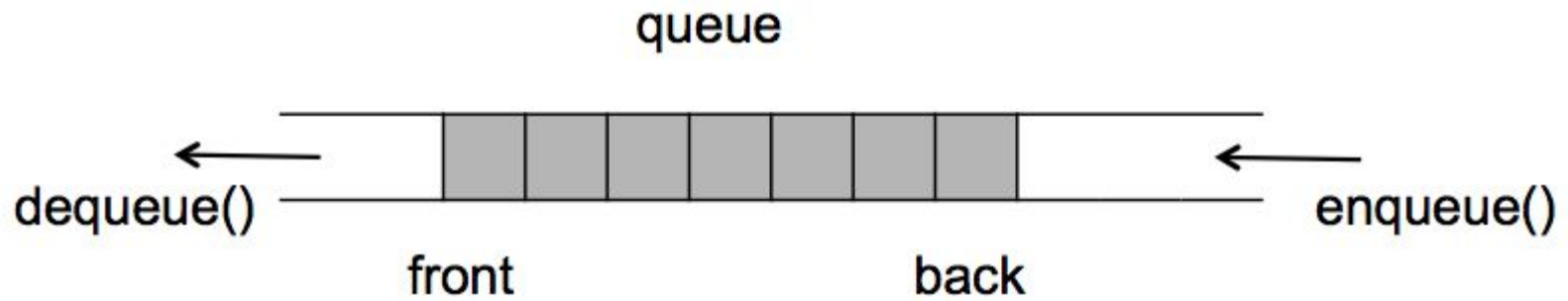
Exerciții

- <https://www.infoarena.ro/problema/nrpits>
- Inversarea unui text
- Problema [parantezelor](#)

Cozi (Queue)

- Sunt structuri de date de tip **FIFO** (**F**irst **I**n **F**irst **O**ut)
- Avem acces la primul și la ultimul element (head & tail / front & back)
- Operații de bază:
 - Push - adăugarea unui element la coadă
 - Pop - eliminarea unui element
- Operații suplimentare:
 - Size() - numărul de elemente
 - isEmpty() - returnează **true** dacă numărul de elemente este exact 0
 - First() - ne spune valoarea de la început fără să o extragă
 - Last() - ne spune valoarea de la sfârșit fără să o extragă

Cozi (Queue)



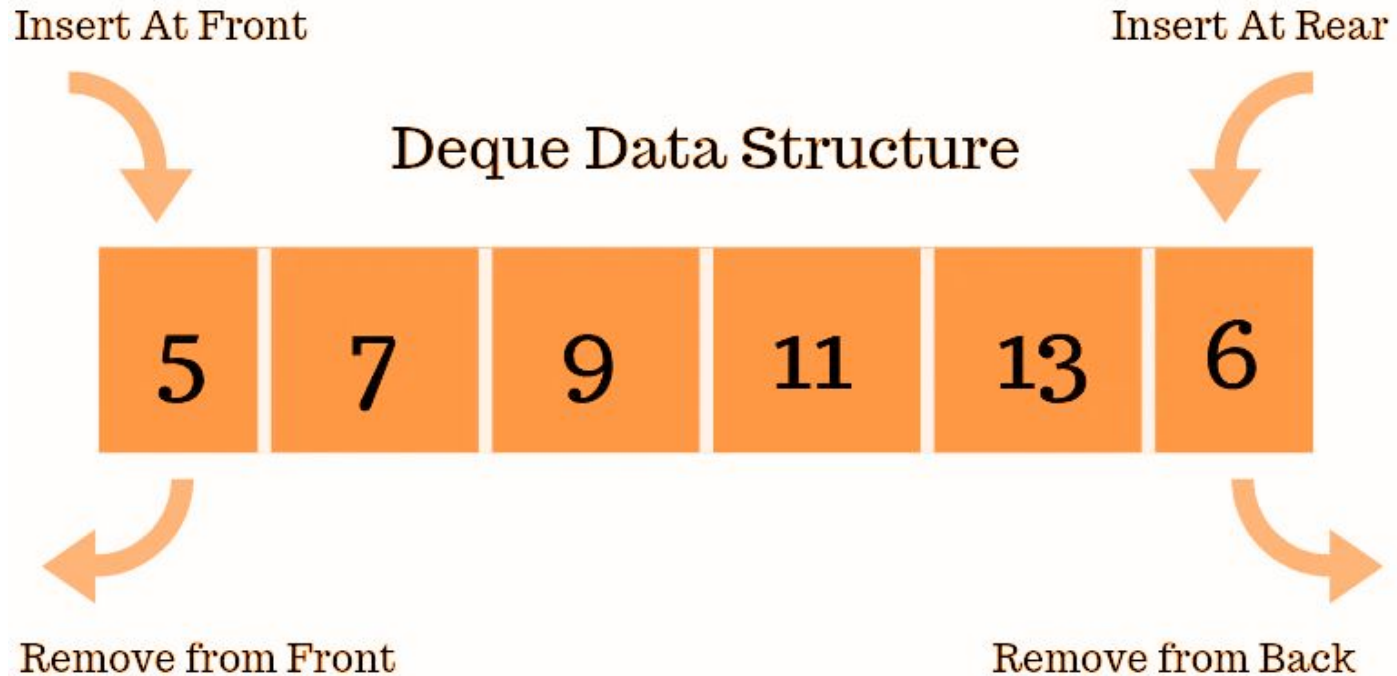
Cozi (Queue)

- Metode de implementare:
 - Coadă ca Vector
 - [Vizualizare](#)
 - [Implementare](#)
 - Coadă ca Listă
 - [Vizualizare](#)
 - [Implementare](#)
 - Coadă în C++ - <https://en.cppreference.com/w/cpp/container/queue>

Deque

- Double ended queue (coadă cu două capete)
- Operații de bază:
 - Push Front
 - Push Back
 - Pop Front
 - Pop Back
- Operații suplimentare
 - Size()
 - Front()
 - Back()
 - isEmpty()

Deque



Deque

- Double ended queue (coadă cu două capete)
- Metode de implementare:
 - Deque ca Listă
 - [Vizualizare](#)
 - [Implementare](#)
 - Deque ca Array
 - [Implementare](#)
 - Deque în C++ - <https://en.cppreference.com/w/cpp/container/deque>

Exerciții

- <https://infoarena.ro/problema/deque>
- Book Pile - <https://codeforces.com/problemsets/acmsguru/problem/99999/271>

Problemă

Se dă un vector cu n elemente și apoi n operații de genul:

- 1 $i\ j \rightarrow$ care este minimul din intervalul $[i, j]$
- 2 $i\ x \rightarrow$ modificați elementul de pe poziția i în x

0	1	2	3	4	5	6	7	8
3	9	2	5	7	34	6	11	8

Problemă

Se dă un vector cu n elemente și apoi n operații de genul:

- 1 i j \rightarrow care este minimul din intervalul $[i, j]$
- 2 i x \rightarrow modificați elementul de pe poziția i în x

Idei?

-

Şmenul lui Batog

Se dă un vector cu n elemente şi apoi n operaţii de genul:

- 1 $i\ j \rightarrow$ care este minimul din intervalul $[i, j]$
- 2 $i\ x \rightarrow$ modificaţi elementul de pe poziţia i în x

Idee:

Împărţim vectorul în zone de lungime L şi calculăm minimul pe fiecare zonă în parte.

0	1	2	3	4	5	6	7	8
3	9	2	5	7	34	6	11	8
2			5			6		

Şmenul lui Batog - SQRT Decomposition

Împărţim vectorul în zone de lungime L şi calculăm minimul pe fiecare zonă în parte.

Linkuri externe:

- [Geeks for geeks](#)
- [CpAlgorithms](#)

Şmenul lui Batog

Se dă un vector cu n elemente şi apoi n operaţii de genul:

- 1 i j \rightarrow care este minimul din intervalul $[i, j]$
- 2 i x \rightarrow modificaţi elementul de pe poziţia i în x

Cum răspundem la 1 0 8; 1 0 4; 1 1 7 ?

0	1	2	3	4	5	6	7	8
3	9	2	5	7	34	6	11	8
2			5			6		

Şmenul lui Batog

Se dă un vector cu n elemente şi apoi n operaţii de genul:

- 1 $i\ j \rightarrow$ care este minimul din intervalul $[i, j]$
- 2 $i\ x \rightarrow$ modificaţi elementul de pe poziţia i în x

Cum răspundem la 1 1 7 ?

0	1	2	3	4	5	6	7	8
3	9	2	5	7	34	6	11	8
2			5			6		

Şmenul lui Batog - Complexitate 1

Pentru query (operația de tip 1):

Împărțim vectorul în **n / L zone** de lungime L

Putem itera aproape complet **2 zone** (de la început și/sau de la final) $\Rightarrow O(2 * L)$

$\Rightarrow O(n/L + 2 * L)$

0	1	2	3	4	5	6	7	8
3	9	2	5	7	34	6	11	8
2			5			6		

Şmenul lui Batog - Complexitate 1

Pentru query (operația de tip 1):

$$O(n/L + 2 * L)$$

Cât trebuie să fie L pentru o complexitate minimă?

- $L = \sqrt{n}$

$$\Rightarrow O(n/\sqrt{n} + 2 * \sqrt{n})$$

$$= O(\sqrt{n} + 2 * \sqrt{n})$$

$$= O(\sqrt{n})$$

Şmenul lui Batog

Se dă un vector cu n elemente şi apoi n operaţii de genul:

- 1 $i\ j \rightarrow$ care este minimul din intervalul $[i, j]$
- 2 $i\ x \rightarrow$ modificaţi elementul de pe poziţia i în x

Cum răspundem la $2\ 2\ 1; \ 2\ 3\ 10\ ?$

0	1	2	3	4	5	6	7	8
3	9	2	5	7	34	6	11	8
2			5			6		

Şmenul lui Batog

Se dă un vector cu n elemente şi apoi n operaţii de genul:

- 1 $i\ j \rightarrow$ care este minimul din intervalul $[i, j]$
- 2 $i\ x \rightarrow$ modificaţi elementul de pe poziţia i în x

Cum răspundem la 2 3 10 ?

0	1	2	3	4	5	6	7	8
3	9	2	5 10	7	34	6	11	8
2			7			6		

Şmenul lui Batog - Complexitate 2

Pentru **update** (operația de tip 2):

Modificăm elementul de pe poziția i

Trebuie să facem update pe zona respectivă (să recalculăm minimul)

⇒ $O(L) = O(\text{sqrt}(n))$

0	1	2	3	4	5	6	7	8
3	9	2	5 10	7	34	6	11	8
2			7			6		

Şmenul lui Batog

Împărţim vectorul în zone de:

- \sqrt{n}
 - $\sqrt{n} / 2$
 - $\sqrt{n} * 2$
 - Variaţiuni
-
- De ce?

Şmenul lui Batog

Împărţim vectorul în zone de:

- $\text{sqrt}(n)$
 - $\text{sqrt}(n) / 2$
 - $\text{sqrt}(n) * 2$
 - Variaţiuni
-
- De ce?
 - În practică, algoritmul poate rula mai rapid pentru valori diferite de $\text{sqrt}(n)$, în funcţie de operaţiile care se fac pe segmente.

Şmenul lui Batog - sortare

Se dă un vector cu n elemente. Sortaţi-l folosind şmenul lui Batog.

0	1	2	3	4	5	6	7	8
3	9	2	5	7	34	6	11	8
2			5			6		

<https://leetcode.com/problems/sort-an-array/>

Complexitate?

- $O(n \sqrt{n})$

Kahoot

Începem cursul 5 (Luni 15 ianuarie ora 14:00) cu **Kahoot dublu!**

Final