

Tad Stiva (Stack)

Ce este o stiva?

O stiva este o structura de date liniara in care elementele se adauga si elimina numai la un singur capat : varful stivei.

Este o structura de tip LIFO - Last In, First Out, adica primul element introdus este primul scos.

Intuitie din viata reala :

O gramada de farfurii puse una peste alta.

Operatii fundamentale :

- push(x) - adauga elementul x in varful stivei
- pop() - elimina elementul din varful stivei
- top() - returneaza elementul din varf fara sa-l elimine
- vida() - verifica daca stiva este goala
- dim() - returneaza numarul de elemente din stiva

Reguli specifice :

- Poti accesa doar varful stivei.
- Nu exista acces "la pozitie" ca in vectori sau liste.
- Nu poti parcurge stiva in orice directie - doar adaugi si elimini din varf.

Implementare :

Pentru implementare se pot folosi vectori, liste sau tablouri statice. Operatiile sunt rapide ($O(1)$ pentru push si pop)

TAD Coadă cu Prioritati (Priority Queue)

Ce este?

O coada cu prioritati este o structura de date in care fiecare element are asociata o prioritate. Cand extragem un element, se scoate cel cu prioritatea cea mai mare, nu cel mai vechi.

Intuitie reala :

Pacientii au prioritate in functie de gravitate, daca altcineva are o urgenta mai mare, va fi tratat inainte.

Operatii fundamentale :

- adauga(e, p) - adauga elementul e cu prioritatea p
- element() - returneaza elementul cu prioritatea cea mai mare
- sterge() - elimina elementul cu prioritate maxima
- vida() - verifica daca coada este goala
- dim() - afiseaza numarul de elemente din coada
- modifica(e, p1, p2) - modifica prioritatea unui element din p1 in p2

Reguli specifice :

- Daca mai multe elemente au aceeasi prioritate, extragerea se face in ordinea adaugarii.

Implementare :

Pentru implementare se poate folosi lista ordonata sau heap-uri binare.

TAD Lista

Ce este?

O lista este o colectie ordonata de elemente care permite :

- inserari si stergeri la orice pozitie
- acces direct la orice element din lista printr-un iterator

Operatii fundamentale :

- `prim()` - returneaza un iterator la primul element
- `adauga(poz, e)` - adauga e dupa pozitia poz
- `sterge(poz)` - elimina elementul de la pozitia poz
- `element(poz)` - returneaza elementul de la pozitia poz
- `cauta(el)` - cauta elementul e si returneaza un iterator la el
- `dim()` - da dimensiunea listei
- `vida()` - verifica daca lista e goala

Despre iteratori :

Un **iterator** este un obiect care "merge" pas cu pas prin lista, metodele lui sunt :

- `valid()` - verifica daca pozitia e valida
- `urmator()` - merge la urmatorul element
- `element()` - returneaza elementul curent

Implementare :

Pentru implementare se va folosi un lant de noduri. Un struct Nod care contine valoarea, pointer catre urmatorul nod si/sau pointer catre precedentul nod.

TAD Lista Ordonata

Ce este?

Este o lista in care elementele sunt pastrate intr-o ordine impusa de o relatie de comparatie (ex : crescator sau descrescator)

Operatii specifice :

- `adauga(e)` - insereaza elementul e in pozitia corecta (pastrand ordinea)
- `sterge(e)` - elimina elementul e daca exista
- `cauta(e)` - cauta si returneaza iteratorul la elementul e
- `prim()` - iterator la primul element (cel "mai mic")
- `vida()` - verifica daca lista este goala

Observatii :

- Relatia de ordine este o functie care defineste ordinea. Ex: $rel(x, y) = x < y$ inseamna ordine crescatoare.
- Nu putem adauga elementul oriunde, trebuie respectata relatia.