Algoritmi in podatkovne strukture 1

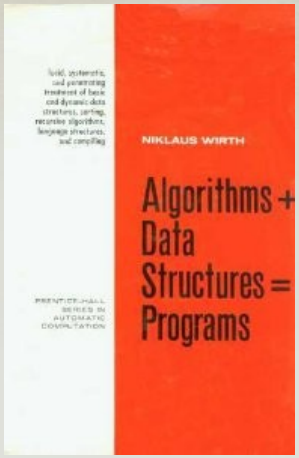
Visokošolski strokovni študij Računalništvo in informatika

Abstraktni podatkovni tipi

Jurij Mihelič, UniLj, FRI

Podatkovni tipi

Niklaus Wirth, 1943 -



Algorithms +
Data Structures
= Programs

Razvil Pascal, Oberon itd.

Software is getting slower more rapidly than hardware becomes faster.

Martin Odersky, 1958 -

Razvil jezik Scala,
…, **generike** v javi.

Podatkovni tipi

- Primer: števila
  - cela števila
    - operacije +, -, *, /, %
    - Java izvedbe: `byte`, `short`, `int`, `long`
    - dvojiški komplement
  - decimalna števila
    - operacije: +, -, *, /
    - Java izvedbe: `float`, `double`
    - IEEE 745 standard

Podatkovni tipi

- Podatkovni tip
  - model za podatke
  - podatek ima ali pripada nekemu tipu
  - pove način uporabe nekega podatka

- Opis podatkovnega tipa
  - množica vrednosti
  - množica operacij
  - predstavitev podatka
  - **razvijalski pogled na podatke**

Podatkovni tipi

- Abstraktni podatkovni tip (ADT – abstract data type)
  - množica vrednosti
  - množica operacij
  - brez implementacije oz. predstavitve podatkov
  - **uporabniški pogled na podatke**
  - analogija iz OOP: vmesnik in razred
  - algebrski pogled na ADT
    - matematična definicija obnašanja tipa
    - podobno algebrskim strukturam (vrednosti, operacije)

Podatkovni tipi

- Objektni opis
  - podatkovna struktura je prejemnik metode

```
ADT s
s.operacija(x)
```

- Klasični opis
  - podatkovna struktura podana kot argument

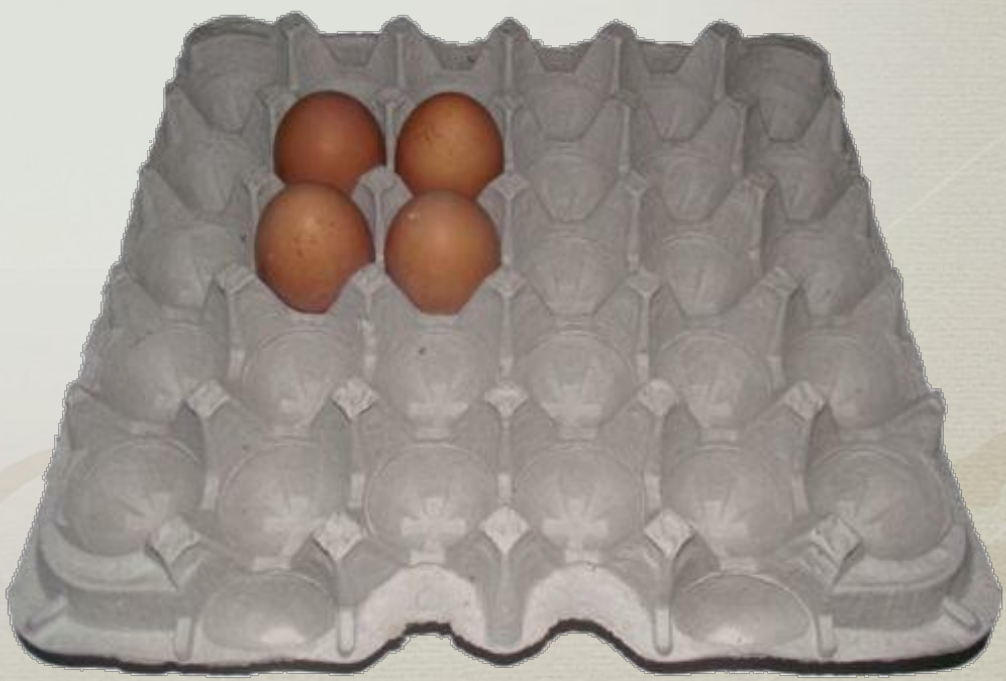
```
ADT s
operacija(s, x)
```

Podatkovni tipi

- Enostavni podatkovni tipi
  - tudi primitivni, atomični, ipd
  - primeri
    - cela števila, števila s plavajočo vejico, znaki
- Sestavljeni podatkovni tipi
  - sestavljeni iz drugih tipov
    - zajemajo njihovo predstavitev (struct)
    - zajemajo njihovo obnašanje (class)
  - primeri
    - kompleksna števila, nizi, zaporedja, strukture, razredi

Podatkovni tipi

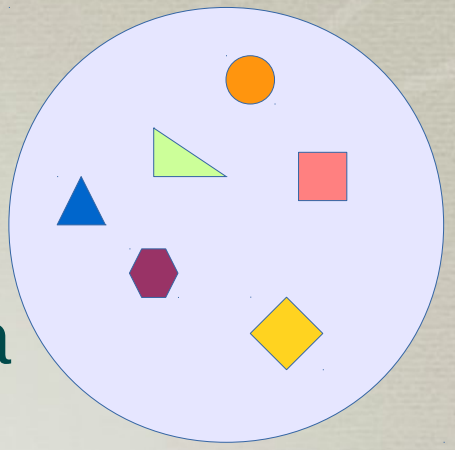
- Podatkovna struktura
  - sestavljen podatkovni tip
  - omogoča učinkovito uporabo podatkov
    - iskanje, dostop, spreminjanje podatkov
    - vključuje predstavitev oz. organizacijo podatkov



Podatkovni tipi

- Osnovni tipi
  - množica
  - vreča
  - sklad
  - vrsta
  - vrsta z dvema koncema
  - vrsta s prednostjo
  - zaporedje
  - slovar

Množica

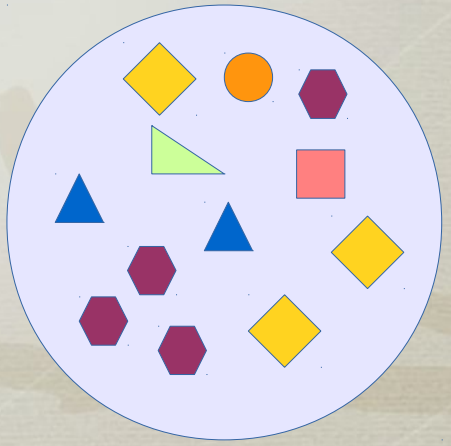


- Množica (*set*)
  - matematično gledano: končna množica
  - vsebuje enolične elemente
  - brez vrstnega reda

Set/Bag

```
find(x)
add(x)
remove(x)
```

- Vreča (*bag, multiset*)
  - kot množica
  - dovoljuje ponavljanje elementov



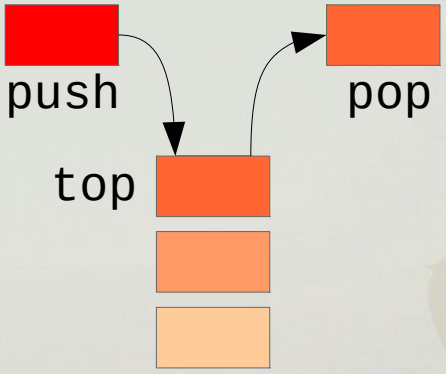
- Urejena množica/vreča
  - vsebuje še operaciji `min()` in `max()`

Sklad

- Sklad (*stack*)
  - LIFO – *last-in, first-out*
  - vrh sklada
  - operaciji `push` in `pop`

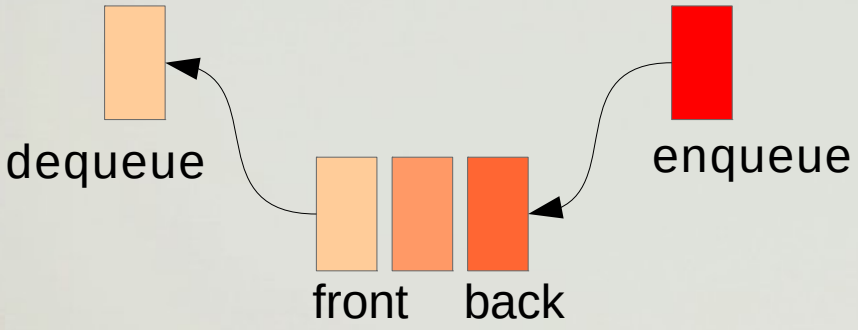
Stack

```
push(x)
pop()

isEmpty()
top()
```



Vrsta

- Vrsta (*queue*)
  - FIFO – *first-in, first-out*
  - sprednji in zadnji konec vrste
  - operaciji enqueue in dequeue



Queue

```
enqueue(x)
dequeue()

isEmpty()
front()
```

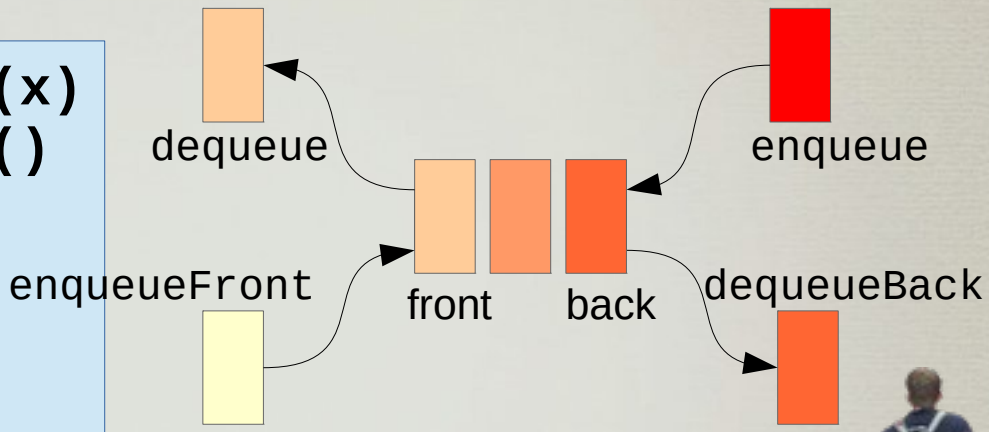
Vrsta z dvema koncema

dvrsta?

- Vrsta z dvema koncema (*deque* ali *dequeue*)
  - dodajanje in odvzemanje spredaj in zadaj

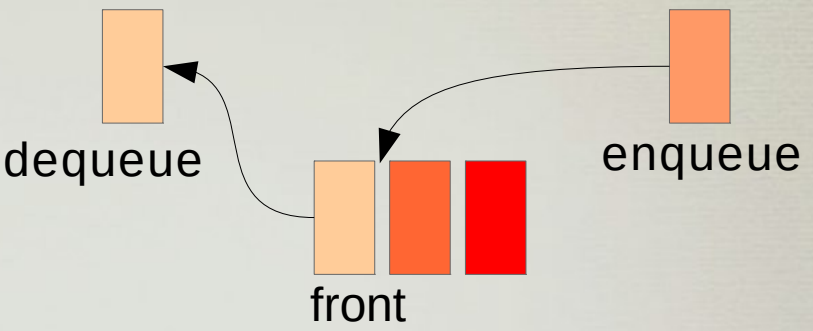
Deque

```
enqueue(x) = enqueueBack(x)
dequeue() = dequeueFront()
enqueueFront(x)
dequeueBack()

isEmpty()
front()
back()
```



Vrsta s prednostjo

- Vrsta s prednostjo (*priority queue*)
  - odvzemanje spredaj
  - dodajanje s prednostjo



PriorityQueue

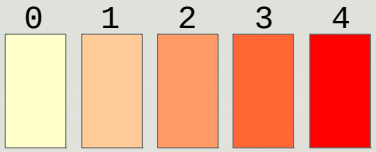
```
enqueue(p, x)
dequeue()

front()
```

Zaporedje

- Zaporedje (*sequence*)
  - naključni dostop: `get(i)`, `set(i, x)`
  - vstavljanje na pozicijo: `insert(i, x)`
  - brisanje na dani poziciji: `delete(i)`

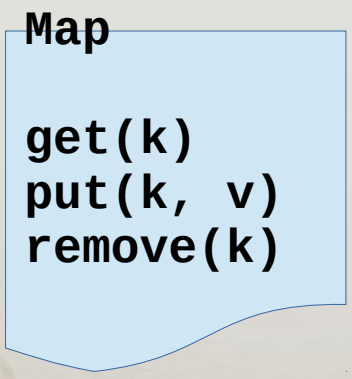
Sequence

```
get(i)
set(i, x)
insert(i, x)
delete(i)
```

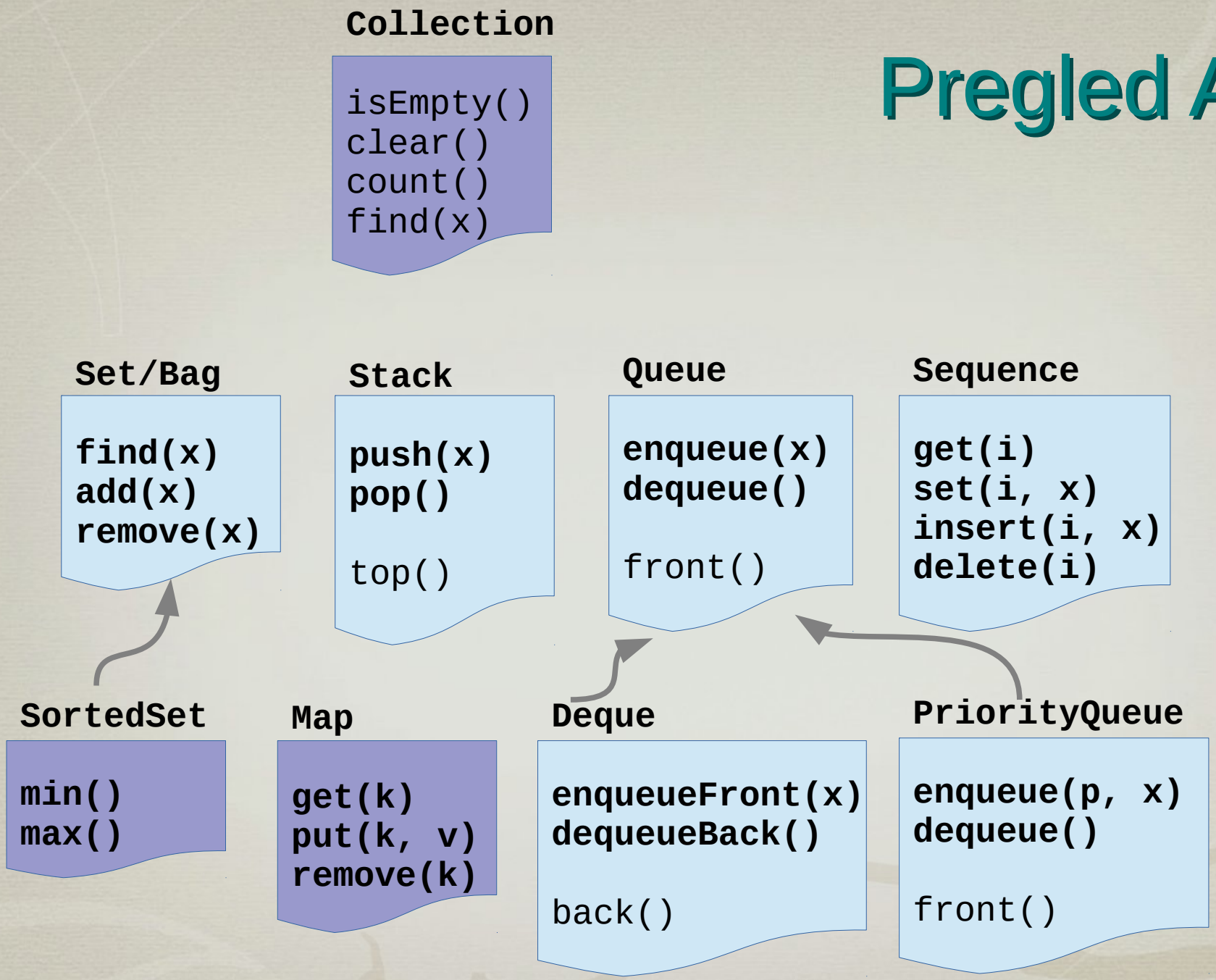


Slovar

- Slovar (*dictionary, map*)
  - podoben množici
  - dostop do elementov preko ključa
  - dostop: `get(k)`, `put(k, v)`
  - odstranjevanje: `remove(k)`



Pregled ADT


Collection
isEmpty()
clear()
count()
find(x)
Set/Bag
find(x)
add(x)
remove(x)
Stack
push(x)
pop()
top()
Queue
enqueue(x)
dequeue()
front()
Sequence
get(i)
set(i, x)
insert(i, x)
delete(i)
SortedSet
min()
max()
Map
get(k)
put(k, v)
remove(k)
Deque
enqueueFront(x)
dequeueBack()
back()
PriorityQueue
enqueue(p, x)
dequeue()
front()

Algebrski pogled

- Algebrske strukture
  - brez (dvojiške) operacije
    - npr. množica, množica z unarno operacijo
  - z eno operacijo +
    - grupoid (magma), semigrupa, monoid, grupa, ...
  - z dvema operacijama + in *
    - polkolobar, kolobar, obseg, algebra
  - več operacij?

Algebrski pogled

- *Nat* =
  - types: Nat
  - operations: $0: \rightarrow \mathrm{Nat}$
    $\mathrm{succ}: \mathrm{Nat} \rightarrow \mathrm{Nat}$
    $\mathrm{add}: \mathrm{Nat}, \mathrm{Nat} \rightarrow \mathrm{Nat}$
  - equations: $n, m \in \mathrm{Nat}$
    $\mathrm{add}(n, 0) = n$
    $\mathrm{add}(n, \mathrm{succ}(m)) = \mathrm{succ}(\mathrm{add}(n, m))$

Algebrski pogled

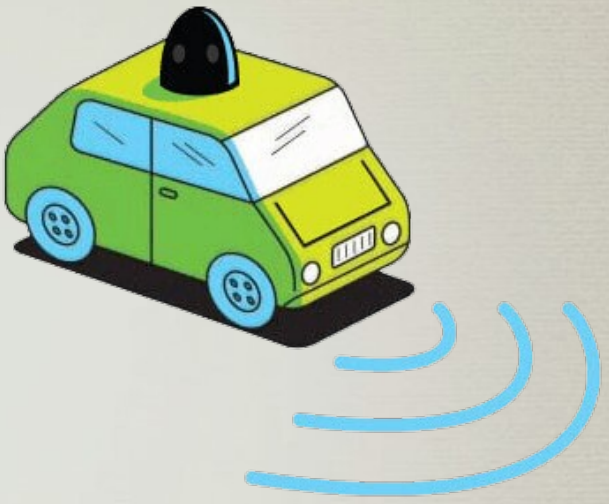
- *Int* =
  - types: Int
  - operations:
    $0: \rightarrow \mathrm{Int}$
    $\mathrm{succ}: \mathrm{Int} \rightarrow \mathrm{Int}$
    $\mathrm{pred}: \mathrm{Int} \rightarrow \mathrm{Int}$
    $\mathrm{add}: \mathrm{Int}, \mathrm{Int} \rightarrow \mathrm{Int}$
  - equations:
    $n, m \in \mathrm{Int}$
    $\mathrm{succ}(\mathrm{pred}(n)) = n$
    $\mathrm{pred}(\mathrm{succ}(n)) = n$
    $\mathrm{add}(n, 0) = n$
    $\mathrm{add}(n, \mathrm{succ}(m)) = \mathrm{succ}(\mathrm{add}(n, m))$
    $\mathrm{add}(n, \mathrm{pred}(m)) = \mathrm{pred}(\mathrm{add}(n, m))$

Algebrski pogled

- *Stack* =
  - types: Item = Int ∪ Nat
    Stack = Item* ∪ {error}
  - operations: empty: → Stack
    push: Item, Stack → Stack
    pop: Stack → Stack
    top: Stack → Item
  - equations: pop(push(x, s)) = s
    top(push(x, s)) = x
    pop(empty) = empty
    top(empty) = error

Algebrski pogled

- Zakaj se sploh truditi z *matematiko*?
  - ker omogoča izredno jasen in natančen pogled na podatkovne strukture
  - ker omogoča **dokaz pravilnosti** delovanja



- Kdaj resnično rabimo pravilnost?
  - preprečevanje katastrof
  - življenjsko kritične aplikacije
    - jedrske elektrarne, avtonomna vožnja
  - varnostno kritične aplikacije
    - bančne aplikacije