PONAVLJANJE GRADIVA I PRIPREMA ZA ISPIT

Apstraktne strukture podataka

Stog, red, stablo

OSNOVNO

- klasa deque
  - modul collections
  - dvostrani red – možemo lako dodavati i brisati elemente s oba kraja
  - metode: d.append(x), d.appendleft(x), d.clear(), d.extend(L), d.extendleft(L), d.pop(), d.popleft()
- klasa stog (Stack)
  - zadnji dodan element će prvi izaći (LIFO)
  - metode: s.push(x), s.pop(), s.isEmpty()
- klasa red (Queue)
  - prvi koji je stavljen u red će prvi izaći (FIFO)
  - metode: q.enqueue(x), q.dequeue(), q.isEmpty()

1. OBJASTITE KRATKO ŠTO ĆE SE DOGODITI U SVAKOJ LINIJI SLJEDEĆEG KODA:

```
from collections import deque
d = deque()
d.append('a')
d.append(4)
d.appendleft(7)
d.append('B')
print(d)
d.popleft()
print(d)
d.extend([4, 3, "F"])
print(d)
print(len(d))
print(d[1], d[0])
```

KLASA STACK

```
from collections import deque

class Stack(deque):
    def __init__(self):
        super().__init__()

    def push(self, v):
        self.append(v)

    def pop(self):
        if not self.isEmpty():
            return super().pop()
        return None

    def isEmpty(self):
        return len(self) == 0
```

Klasa Queue

```
from collections import deque
class Queue(deque):
    def __init__(self):
        super().__init__()

    def enqueue(self, t):
        self.append(t)

    def dequeue(self):
        return self.popleft()

    def isEmpty(self):
        return len(self) == 0
```

2. OBJASTITE KRATKO ŠTO ĆE SE DOGODITI U SVAKOJ LINIJI SLJEDEĆEG KODA:

```
from Stack import *
s = Stack()
print(s.isEmpty())
s.push(2)
s.push(1)
s.push(9)
print(s)
s.pop()
s.push(3)
s.push(12)
s.pop()
print(s)
print(s.pop())
```

3. OBJASTITE KRATKO ŠTO ĆE SE DOGODITI U SVAKOJ LINIJI SLJEDEĆEG KODA:

```
from Queue import *
q = Queue()
q.enqueue("A")
q.enqueue("X")
q.enqueue("C")
print(q.isEmpty())
q.enqueue("Y")
q.enqueue("Z")
q.dequeue()
q.dequeue()
print(q)
q.enqueue("B")
q.dequeue()
q.dequeue()
print(q)
```

PISANJE ARITMETIČKIH IZRAZA (INFIX, PREFIX, POSTFIX)

4. Sljedeće aritmetičke izraze zapisane u infiksnim načinu zapišimo u prefix-notaciji i postfix-notaciji:

a) (4 + 2) / 7 – 3

b) (4 – 1) *(3 + 2) – 4

c) ((10 – 9) / 2 – 1)*10

5. Sljedeće aritmetičke izraze zapisane u postfix-notaciji zapišimo u infix-notaciji

a) 5 4 – 3 * 2 -

b) 3 1 – 7 3 + / 4 *

6. Sljedeće aritmetičke izraze zapisane u prefix- notaciji zapišimo u infix-notaciji.

a) + * 9 – 4 1 7

b) – * 3 5 / + 1 4 2

STABLO

- razgranata struktura
- stablo se sastoji od **čvorova**
- **korijen** – prvi čvor
- čvor koji ima jedno ili dvoje djece – **roditelj** toj djeci (lijevo i desno dijete)
- djeca istog čvora – **susjedi**
- čvorovi koji nemaju djece – **listovi**
- **binarno stablo –** svaki čvor ima najviše dva djeteta
- **unutarnji čvor** – ima najmanje jedno dijete
- **duljina puta od korijena** do nekog čvora nazivamo **razina** čvora n (korijen je razina 0)
- **dubina stabla** je udaljenost od korijena do najudaljenijeg čvora
- **stupanj** čvora je broj djece tog čvora
- binarno stablo reći ćemo da je **puno** (engl. full) ako svaki čvor bez listova ima stupanj 2
- **potpuno** (engl. complete) binarno stabl na svim razinama k osim zadnje ima 2^k čvorova

OBILASCI STABLA

- preorder –korijen, lijevo podstablo pa desno podstablo
- inorder – prvo lijevo podstablo, zatim korijen pa desno podstablo
- postorder – prvo lijevo podstablo, zatim desno podstablo i na kraju korijen

7. Za binarno stablo na slici odredi:

a) Koji je korijen stabla?

b) Kolika je dubina nacrtanog stabla?

c) Koja su djeca čvora D, a koja čvora H?

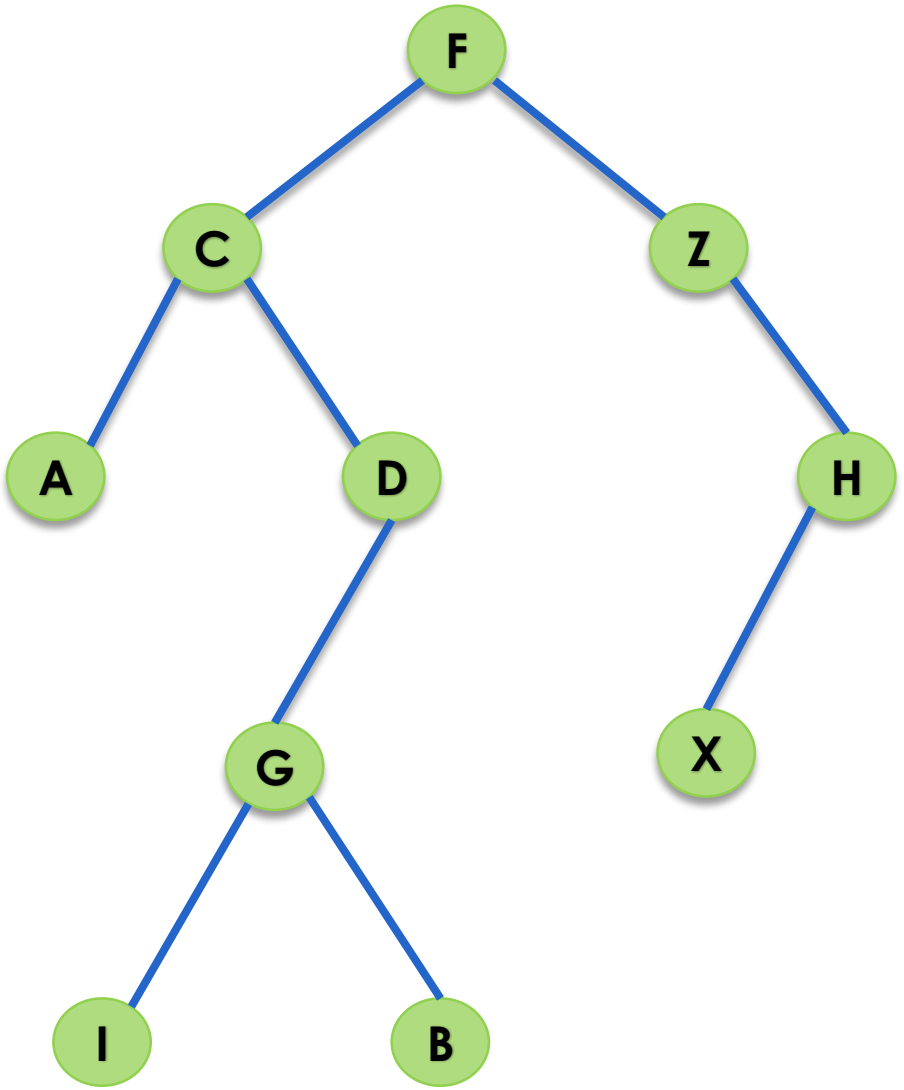
d) Na kojoj se razini nalazi čvor B, a na kojoj razini čvor H?

e) Koji je roditelj čvora F, a koji čvora A?

f) Ispiši obilazak vrhova stabla preorder metodom.

g) Ispiši obilazak vrhova stabla inorder metodom.

h) Ispiši obilazak vrhova stabla postorder metodom.



8. Kreirajmo binarno stablo za sljedeće izraze:

a) (a – b) /(d + c)

b) (a + b * c) – d/e

- obiđimo stablo i njegove elemente preorder i postorder metodom

ZADATAK 5.

9. Kreirajmo binarno stablo ako nam je zadan inorder obilazak stabla: DFHBCAE i preorder obilazak: ABFDHCE.

10. Kreirajmo binarno stablo ako nam je zadan inorder obilazak stabla: ACBDFE i postorder obilazak: BFDCEA.