



IPAQ PETA

V. GIMNAZIJA ZAGREB

GIMNAZIJA VUKOVAR

SREDNJA ŠKOLA LOVRE MONTIJA KNIN

SREDNJA ŠKOLA PAKRAC

GIMNAZIJA METKOVIĆ

PMF ZAGREB

Graf

ULAGANJE U BUDUĆNOST



PROJEKT
SUFINANCIRA
EUROPSKA UNIJA

Ova publikacija je izrađena uz pomoć Europske unije. Sadržaj publikacije je u isključivoj nadležnosti V. gimnazije te se ni na koji način ne može smatrati da odražava stajališta Europske unije.

Europsku uniju čini 28 država članica koje su odlučile postupno povezivati svoja znanja, resurse i sudbine. Tijekom 50-godišnjeg razdoblja proširivanja, zajedno su izgradile područje stabilnosti, demokracije i održivog razvoja, zadržavajući pritom kulturnu raznolikost, toleranciju i slobode pojedinaca. Europska unija je posvećena dijeljenju svojih postignuća i vrijednosti s državama i narodima izvan svojih granica.

Graf

- graf – skup čvorova od kojih su neki međusobno povezani granama
- dimenzija grafa – broj čvorova
- težinski graf – grane imaju težinu
- (ne)usmjereni graf – grane nemaju/imaju smjer
- put – niz čvorova povezanih granama (bez ponavljanja grana)
- duljina puta – broj grana na putu
- ciklus – put s istim početkom i završetkom
- aciklički graf (digraf) – graf bez ciklusa

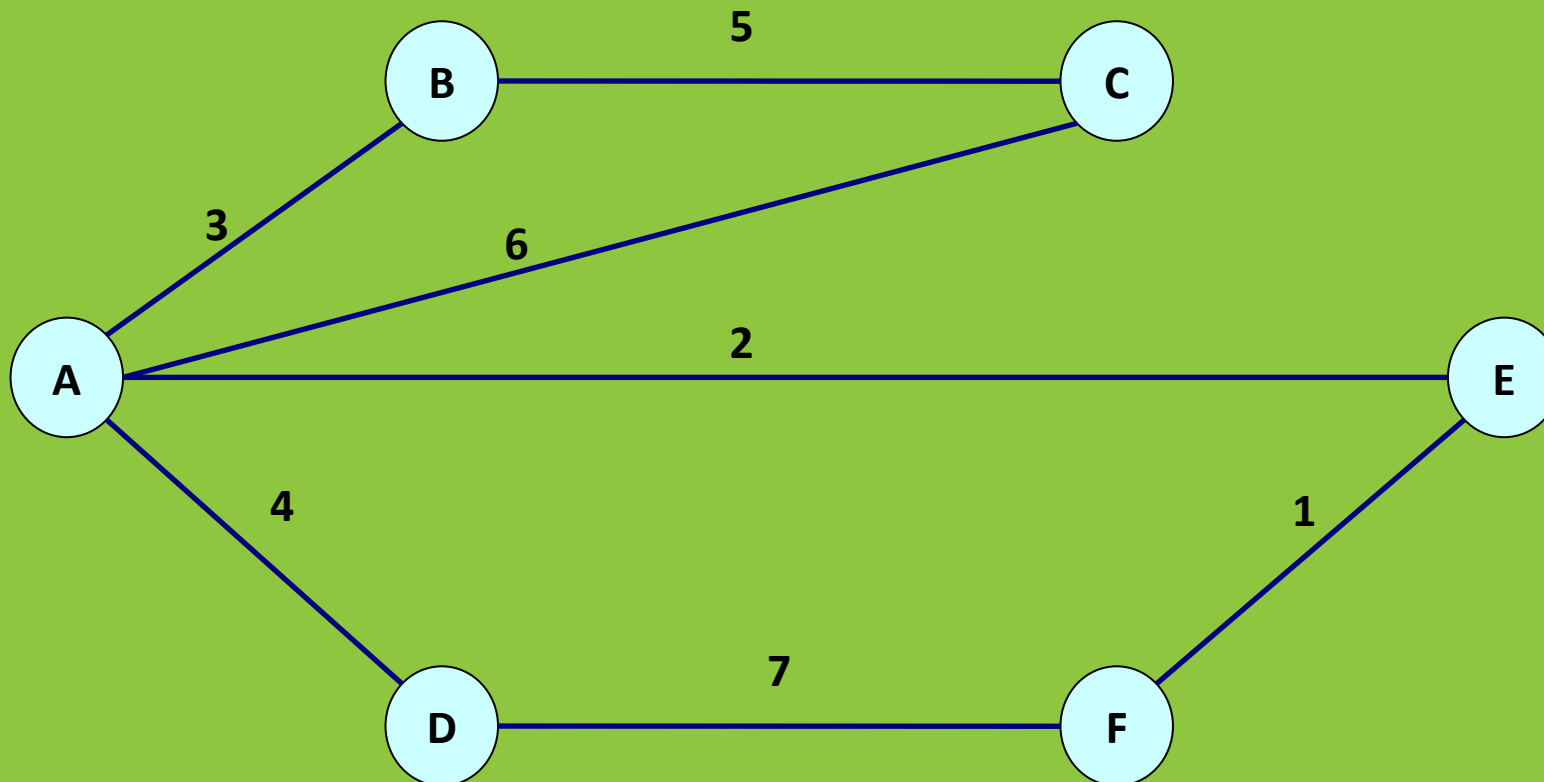
ULAGANJE U BUDUĆNOST



PROJEKT
SUFINANCIRA
EUROPSKA UNIJA



Graf - primjer 8



1. Odredi dimenziju grafa
2. odredi broj bridova
3. ispiši sve putove duljine 3 iz čvora E
4. ispiši sve cikluse

1. **6**
2. **7**
3. **EABC, EACB, EADF, EFDA**
4. **ABCA, ADFEA**

ULAGANJE U BUDUĆNOST



PROJEKT
SUFINANCIRA
EUROPSKA UNIJA



Graf



ULAGANJE U BUDUĆNOST



Pohranjivanje grafa u računalu

- lista susjedstva
 - za svaki čvor u pripadnoj listi pišu redni brojevi svih njegovih susjeda
- matrica susjedstva
 - u i-tom redu matrice u stupcu j piše 1 ako postoji grana od čvora i do čvora j, inače piše 0
- težinska matrica
 - u i-tom redu matrice u stupcu j piše težina grane. Ukoliko grana ne postoji piše beskonačno (∞)

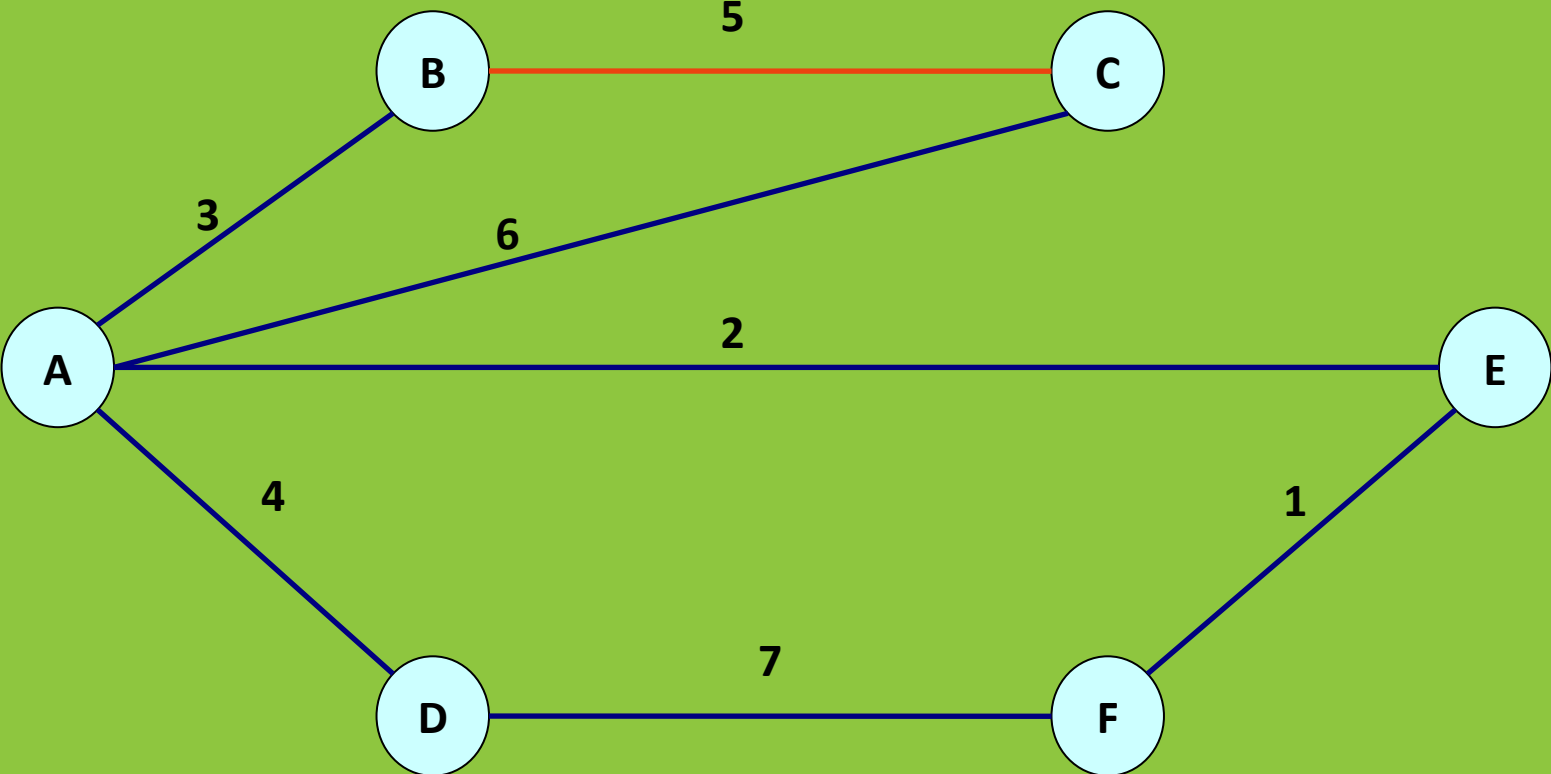
ULAGANJE U BUDUĆNOST



PROJEKT
SUFINANCIRA
EUROPSKA UNIJA



Pohranjivanje grafa u računalu - primjer 9



Lista susjedstva:

- [1, 2, 3, 4], [0, 2],
- [0, 1], [0, 5], [0, 5],
- [3, 4]

Matrica susjedstva:

0	1	1	1	1	0
1	0	1	0	0	0
1	1	0	0	0	0
1	0	0	0	0	1
1	0	0	0	0	1
0	0	0	1	1	0

Težinska matrica:

∞	3	6	4	2	∞
3	∞	5	∞	∞	∞
6	5	∞	∞	∞	∞
4	∞	∞	∞	∞	7
2	∞	∞	∞	∞	1
∞	∞	∞	7	1	∞

ULAGANJE U BUDUĆNOST



PROJEKT
SUFINANCIRA
EUROPSKA UNIJA



Klasa Matrica

```
class Matrica:
    def __init__(self, n, m = 0):
        self.n = n
        if m == 0:
            self.m = n
        else:
            self.m = m
        self.mat = dict()
        return

    def __setitem__(self, ključ, vrijednost):
        self.mat[ključ] = vrijednost
        return

    def __getitem__(self, ključ):
        return self.mat.get(ključ, 0)

    def __repr__(self):
        s = ''
        for i in range(self.n):
            for j in range(self.m):
                s = s + '{0:3}'.format(self[i, j])
            s = s + '\n'
        return s
```

ULAGANJE U BUDUĆNOST



PROJEKT
SUFINANCIRA
EUROPSKA UNIJA



Klasa Matrica - primjeri

```
>>> a = Matrica(2, 3)
```

```
>>> a[0, 0] = 3
```

```
>>> a[0, 1]
```

```
0
```

```
>>> a[0, 2] = 2
```

```
>>> a[1, 2] = 4
```

```
>>> a
```

```
3 0 2
```

```
0 0 4
```

ULAGANJE U BUDUĆNOST



PROJEKT
SUFINANCIRA
EUROPSKA UNIJA



Pretraživanje grafa

- kako programski doći do svih čvorova grafa?
 - širinsko pretraživanje
 - dubinsko pretraživanje

ULAGANJE U BUDUĆNOST



PROJEKT
SUFINANCIRA
EUROPSKA UNIJA



Širinsko pretraživanje grafa (**B**readth **F**irst **S**earch)

- odaberemo neki čvor
- u prvoj iteraciji u rječnik stavimo sve susjede prvog čvora
- u svakoj sljedećoj iteraciji uzmemo iz rječnika prvi element s reda, proglasimo ga obištenim te na red postavimo sve njegove susjede koji još nisu u redu i koji još nisu obišteni

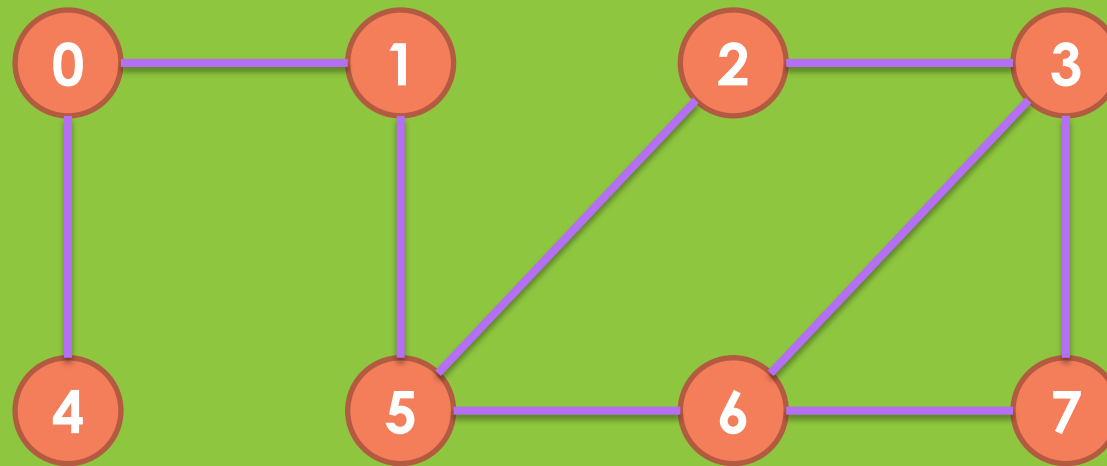
ULAGANJE U BUDUĆNOST



PROJEKT
SUFINANCIRA
EUROPSKA UNIJA



Širinsko pretraživanje grafa - primjer



ULAGANJE U BUDUĆNOST



PROJEKT
SUFINANCIRA
EUROPSKA UNIJA



Širinsko pretraživanje grafa - implementacija

```
def bfs(a, v):  
    posjeceni = {v}  
    obrada = []  
    r = Queue()  
    r.enqueue(v)  
    while not r.isEmpty():  
        v = r.dequeue()  
        obrada.append(v)  
        for i in range(a.m):  
            if a[v, i] == 1 and not i in posjeceni:  
                r.enqueue(i)  
                posjeceni |= {i}  
    return obrada
```

ULAGANJE U BUDUĆNOST



PROJEKT
SUFINANCIRA
EUROPSKA UNIJA



Dubinsko pretraživanje grafa (Deep First Search)

- odaberemo neki čvor (c_1)
- posjetimo njegovog prvog (c_2) susjeda
- posjetimo prvog susjeda čvora (c_2)
- ...
- kada više ne možemo ići u dubinu vraćamo se natrag sve dok ne dođemo do čvora koji ima susjeda u kojem još nismo bili

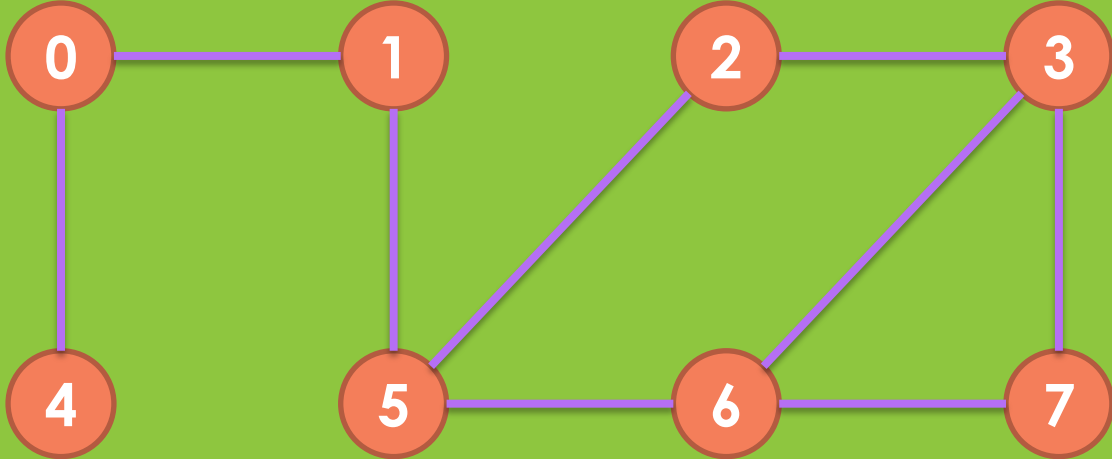
ULAGANJE U BUDUĆNOST



PROJEKT
SUFINANCIRA
EUROPSKA UNIJA



Dubinsko pretraživanje grafa - primjer



ULAGANJE U BUDUĆNOST



PROJEKT
SUFINANCIRA
EUROPSKA UNIJA



Dubinsko pretraživanje grafa - implementacija

```
def dfs(a, v):  
    posjeceni = {v}  
    obrada = [v]  
    s = Stack()  
    s.push(v)  
    while not s.isEmpty():  
        v = s.pop()  
        i = 0  
        while i < a.m and a[v, i] == 0 or i in posjeceni:  
            i += 1  
        if i < a.m:  
            posjeceni |= {i}; s.push(v); s.push(i)  
            obrada.append(i)  
    return obrada
```

ULAGANJE U BUDUĆNOST



PROJEKT
SUFINANCIRA
EUROPSKA UNIJA

