

**SVEUČILIŠTE U ZAGREBU**  
**FAKULTET ORGANIZACIJE I INFORMATIKE**  
**V A R A Ž D I N**

**Patrick Pok**

**Karlo Previšić**

**Marko Radović**

**Društveni izbori i preferencijalno glasanje**

**PROJEKTNI RAD IZ KOLEGIJA DISKRETNE STRUKTURE S  
TEORIJOM GRAFOVA**

**Varaždin, 2024.**

**SVEUČILIŠTE U ZAGREBU**  
**FAKULTET ORGANIZACIJE I INFORMATIKE**  
**V A R A Ž D I N**

**Patrick Pok – Matični broj: 0016149374**

**Karlo Previšić – Matični broj: 0016150996**

**Marko Radović – Matični broj: 0016151049**

**Studij: Organizacija poslovnih sustava**

**Društveni izbori i preferencijalno glasanje**

**PROJEKTNI RAD IZ KOLEGIJA DISKRETNE STRUKTURE S TEORIJOM GRAFOVA**

**Mentorica:**

**Doc. dr. sc. Marcel Maretić**

**Varaždin, siječanj 2024.**

### **Izjava o izvornosti**

Izjavljujemo da naš projektni rad izvorni rezultat našeg rada te da se u izradi istoga nismo koristili drugim izvorima osim onima koji su u njemu navedeni. Za izradu rada su korištene etički prikladne i prihvatljive metode i tehnike rada.

*Autori potvrdili prihvaćanjem odredbi u sustavu FOI-radovi*

---

# Sadržaj

Sadržaj.....	iv
1. Strategija rješavanja .....	5
2. Preferencijalno glasanje.....	5
3. Vizualizacija preferencijalnih veza kandidata .....	6
4. Parcijalne funkcije društvenih izbora .....	7
4.1. Bordina metoda.....	7
4.1.1. Zadatak - Bordina metoda .....	7
4.2. Condorcetova metoda.....	8
4.2.1. Zadatak - Condorcetova metoda .....	8
4.3. Većinska metoda .....	9
4.3.1. Zadatak – Većinska metoda .....	9
5. Rubni slučajevi.....	10
6. Taktičko glasanje .....	10
7. Implementacija.....	11
7.1. Rješenje programa.....	13
8. Arrowljev teorem mogućnosti .....	15
Popis literature .....	16
Popis slika .....	17
Popis tablica .....	17

# 1. Strategija rješavanja

Naša strategija za rješavanje ovog projekta ima nekoliko ključnih koraka. Početni cilj nam je stjecanje razumijevanja preferencijalnog glasanja, točnije metode za rješavanje mogućih problema. Metode koje će se koristiti su Bordina, Condorcetova i većinska za koju će biti pokazani sami primjeri kako bi se bolje razumjele i naučile same metode. Istražit će se literatura i načini rješavanja sličnih zadataka kako bi se steklo veće znanje. Ponovit ćemo teoriju grafova kako bismo razumjeli grafički prikaz preferencijalnog glasanja. Istražit ćemo dodatno literaturu kako bismo shvatiti rubne slučajeve i taktičko glasanje. Za implementaciju rješenja prvo ćemo istražiti posebno biblioteke poput „Counter“ koje će olakšati izradu. Strategija za implementaciju bit će orijentirana od implementacije unosa podataka, Bordine metode, većinske metode, Condorcetove metode i vizualizacije zadataka. Za implementaciju će se koristiti već postojeće znanje rada u Pythonu i biblioteke, a navedeni redoslijed implementacije omogućit će lakšu i logičnu izradu projektnog zadatka.

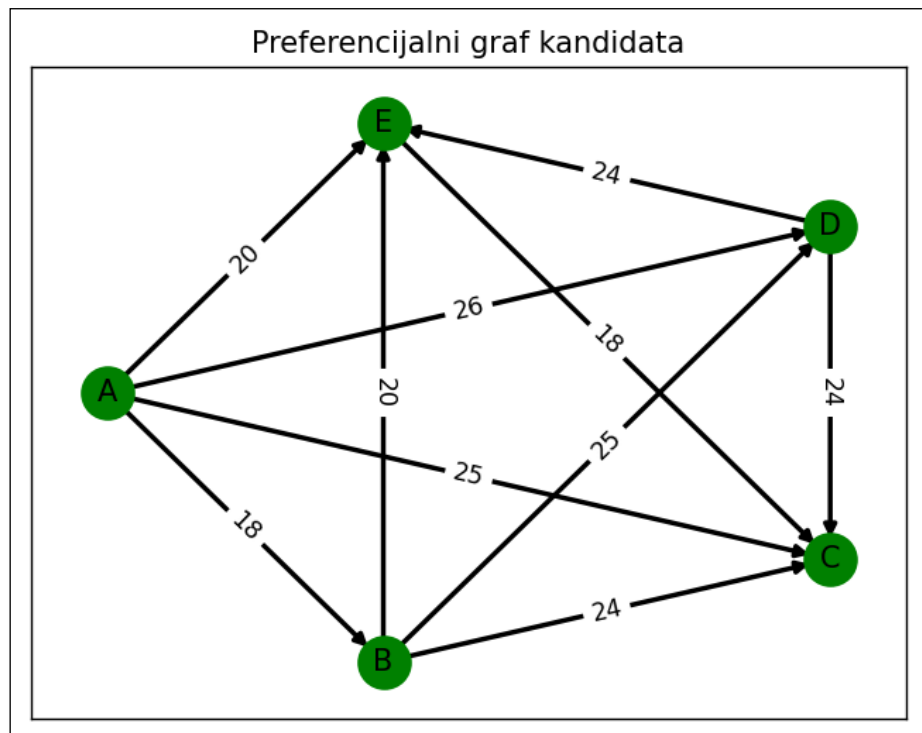
## 2. Preferencijalno glasanje

Agregacija preferencijalnih glasova je složen proces koji uključuje prikupljanje i kombiniranje glasova od pojedinaca kako bi se odabrao pobjednik ili donijele određene odluke. Ovaj proces često postaje problematičan, posebno kada se koriste preferencijalni sustavi glasanja, jer se moraju uzeti u obzir individualne preferencije svakog glasača. U izborima s preferencijalnim glasanjem, birači rangiraju kandidate prema redoslijedu preferencija, pri čemu je "1" najviši (najpoželjniji) rang. Na primjer, ako postoje 3 kandidata, glasač bira 1 pored svog prvog izbora, 2 pored svog drugog izbora i 3 pored svog trećeg izbora. Da bi se smatrao važećim glasačkim listićem, predani glasački listić mora ispunjavati sljedeće uvjete (ElectionBuddy, bez dat.):

- Za svako mjesto mora biti rangiran najmanje jedan kandidat. U slučaju da je suzdržanost dopuštena, a birač se suzdržao od glasanja za neku poziciju, glasački listić može se smatrati važećim za bilo koju drugu poziciju u izborima.
- Dva kandidata ne mogu imati isti rang ili je glasački listić nevažeći (samo za tu poziciju).
- Svaki kandidat može dobiti bilo koji rang jer se broji relativni rang. U ovom slučaju, zadnji kandidat će biti prvi izbor za ovog glasača (budući da je "3." najviši rang označen na glasačkom listiću), zatim treći kandidat, a zatim četvrti kandidat.

### 3. Vizualizacija preferencijalnih veza kandidata

Koristeći biblioteku NetworkX u programskom jeziku Python implementiran je graf za modeliranje preferencijalnih veza između kandidata. Svaki kandidat predstavljen je kao vrh grafa, a bridovi između kandidata odražavaju preferencijalne veze birača. Graf je usmjeren od kandidata koji je dobio više glasova od drugoga u međusobnoj usporedbi, a težina brida predstavlja broj glasova kandidata koji je pobijedio (snagu preferencije). Na primjer, ako je kandidat A pobijedio nad kandidatom B s 25 glasova, tada će brid biti usmjeren od A prema B s težinom brida postavljenom na 25. Ova implementacija omogućuje jasno modeliranje te analizu preferencijalnih odnosa među kandidatima. Na ovom primjeru možemo vidjeti da se navedeni zadatak povezuje s teorijom kolegija, odnosno teorijom grafova.



Slika 1: Preferencijalni graf kandidata (autorski rad)

Slika 1 prikazuje preferencijalni graf kandidata s podacima iz zadatka Condorcetove metode (3.2.1). Implementacija navedenog grafa je prikazana u 6. poglavlju u sklopu Condorcetove metode. Graf može poslužiti da se lakše analiziraju, uvide i iščitaju različiti podaci. Tako na primjer, Vrh A (kandidat) ima najviše izlaznih bridova, odnosno svi su mi izlazni bridovi što ukazuje da je kandidat A pobjeđivao u svim usporedbama s ostalim kandidatima. S druge strane, Vrh C (kandidat) ima sve ulazne bridove, što znači da su svu drugi kandidati pobjeđivali u usporedbama s kandidatom C. Također se može primijetiti kako graf nema ciklusa.

## 4. Parcijalne funkcije društvenih izbora

### 4.1. Bordina metoda

Bordina metoda, nazvana po francuskom matematičaru Jean-Charlesu de Bordi, predstavlja sustav preferencijalnog glasanja koji se koristi prilikom izbora. Bordina metoda je skup pravila glasanja u kojem svaki kandidat dobiva bodove prema njegovom rangiranju na glasačkom listiću. U osnovnoj verziji, najniže rangirani kandidat dobiva 0 bodova, a svaki sljedeći dobiva jedan bod više, sve do najbolje rangiranog kandidata koji prima  $n - 1$  bodova, gdje je  $n$  ukupan broj kandidata. Kandidat ili opcija s najviše bodova nakon prebrojavanja postaje pobjednik (Wikipedia A, 2001).

Metoda ima cilj izabrati opcije ili kandidate koji su široko prihvatljivi, umjesto da se usredotoči samo na većinske preferencije. Stoga se često opisuje kao sustav glasanja temeljen na konsenzusu, čime se naglašava nastojanje da se postigne opća podrška, a ne samo podrška većine (Wikipedia A, 2001).

#### 4.1.1. Zadatak - Bordina metoda

Pretpostavimo da imamo 5 kandidata (A, B, C, D, E) i 20 birača. Svaki birač rangira kandidate prema svojim preferencijama. Birači su postavili kandidate u sljedećem redoslijedu.

Birači	8	5	4	2	1
1. Izbor	C	D	B	E	A
2. Izbor	E	A	C	D	B
3. Izbor	B	E	A	B	D
4. Izbor	D	C	E	A	C
5. Izbor	A	B	D	C	E

Tablica 1: Preferencije birača – Bordina metoda (autorski rad)

Broj opcija birača je  $n = 5$ , a bodovi se dodjeljuju prema formuli  $n - i, i = 1, \dots, n$ , odnosno u ovom slučaju 4 bodova za prvi izbor, 3 boda za drugi izbor i tako dalje. Nakon što su određene preferencije birača, bodovi za izbore se mogu izračunati. Postupak izračuna prikazan je u *Tablici 2*.

A	B	C	D	E
$8*0 = 0$	$8*2 = 16$	$8*4 = 32$	$8*1 = 8$	$8*3 = 24$
$5*3 = 15$	$5*0 = 0$	$5*1 = 5$	$5*4 = 20$	$5*2 = 10$
$4*2 = 8$	$4*4 = 16$	$4*3 = 12$	$4*0 = 0$	$4*1 = 4$
$2*1 = 2$	$2*2 = 4$	$2*0 = 0$	$2*3 = 6$	$2*4 = 8$

$1*4 = 4$	$1*3 = 3$	$1*1 = 1$	$1*2 = 2$	$1*0 = 0$
29	39	50	36	46

Tablica 2: Rješenje - Bordina metode (autorski rad)

Izračunom vidimo da kandidat C u ovom slučaju pobjeđuje s 50 glasova iako je bio samo prvi izbor jedne skupine birača.

## 4.2. Condorcetova metoda

Condorcetova metoda još je jedna vrsta glasovanja koja koristi rangirane listiće za izbor jednog kandidata. Condorcetovom metodom bira se kandidat koji pobjeđuje sve ostale kandidate u izborima po paru. Da bi se otkrio pobjednika Condorceta, mora se uzeti u obzir sve izbore u paru. U slučaju dva kandidata A i B, ako je A rangiran više od B na većini glasačkih listića, tada A pobjeđuje B, inače B pobjeđuje A. Kandidat koji na taj način pobjeđuje sve druge kandidate je pobjednik (OpaVote, bez dat.).

Ovo je najlakše vizualizirati pomoću tablice. U rezultatima usporedite kandidata u retku s kandidatom u stupcu. Ako je ćelija zelena, to znači da je kandidat u retku pobijedio kandidata u stupcu. Ako je ćelija crvena, tada je kandidat u retku izgubio od kandidata u stupcu. Brojevi u ćeliji pokazuju vam broj glasačkih listića u kojima je svaki kandidat bio bolje rangiran od drugog (OpaVote, bez dat.).

Pored toga Condorcetova metoda ima jednu komplikaciju. Moguće je da nema kandidata koji pobjeđuje sve ostale kandidate. Na primjer, moguće je da kandidat A pobijedi kandidata B, kandidat B pobijedi kandidata C, a kandidat C pobijedi kandidata A. Dakle, svi su kandidati izgubili od najmanje jednog drugog kandidata. Ovo se ponekad naziva Condorcetov ciklus ili paradoks (OpaVote, bez dat.).

### 4.2.1. Zadatak - Condorcetova metoda

Pretpostavimo da imamo opet 5 kandidata (A, B, C, D, E) i 30 birača koji su grupirani u pet skupina. Tablica preferencija skupina je prikazana *Tablicom 3*.

Birači	13	7	5	4	1
1. Izbor	A	B	C	D	E
2. Izbor	B	A	E	E	A
3. Izbor	D	D	B	A	C
4. Izbor	E	C	A	B	D
5. Izbor	C	E	D	C	B

Tablica 3: Preferencija birača – Condorcetova metoda (autorski rad)



Kada znamo preferencije birača možemo početi s uspoređivanjem. *Tablica 4* prikazuje rezultate usporedbe svaka dva kandidata gdje su zelenom označene pobjede, a crvenom porazi. Iz tablice se može vidjeti da kandidat A pobjeđuje u svim usporedbama, što znači da je kandidat A Condorcetov pobjednik.

	A	B	C	D	E
A	0	18	25	26	20
B	12	0	24	25	20
C	5	6	0	6	12
D	4	5	24	0	24
E	10	10	18	6	0

Tablica 4: Rješenje - Condorcetova metoda (autorski rad)

### 4.3. Većinska metoda

Većinska metoda je izborni proces u kojem je izabran kandidat koji dobije više glasova od bilo kojeg drugog kandidata. Pluralistički sustav ima prednosti u jednostavnom razumijevanju birača, brzom donošenju odluka te praktičnosti i nižim troškovima. Većinska metoda najučinkovitije djeluje u dvostranačkim sustavima. Odluka se donosi na temelju broja prvih mjesta koje svaki kandidat osvoji prema preferencijama birača (The Editors of Encyclopaedia Britannica, 2008).

#### 4.3.1. Zadatak – Većinska metoda

U ovom konkretnom primjeru pretpostavljamo pet kandidata (A, B, C, D, E) i 25 birača. Tablica u nastavku prikazuje preferencije pet skupina birača. Većinska metoda određuje pobjednika na temelju broja glasova samo prvog izbora što znači da je pobjednik kandidat A s 12 glasova.

Birači	12	7	4	1	1
1. Izbor	A	B	C	D	E
2. Izbor	B	A	E	E	A
3. Izbor	D	D	B	A	C
4. Izbor	E	C	A	B	D
5. Izbor	C	E	D	C	B

Tablica 5: Preferencije birača – Većinska metoda (autorski rad)

## 5. Rubni slučajeви

Prema zadanom zadatku „Razmotrite situaciju u kojoj postoji ciklus pobjeda između kandidata (A pobjeđuje B, B pobjeđuje C, C pobjeđuje A, D pobjeđuje E, E pobjeđuje A, itd.)“ ova situacija predstavlja Condorcetov ciklus, koji je paradoks u kontekstu izbornih sustava. U Condorcetovom ciklusu (paradoksu) ne postoji kandidat koji je pobijedio u svim direktnim usporedbama s ostalim kandidatima. U primjeru, gdje kandidat A pobjeđuje B, B pobjeđuje C, C pobjeđuje A nema pobjednika izbora. Ovakve situacije stvaraju nedefiniranost u određivanju apsolutnog pobjednika prema Condorcetovom pravilu, što je paradoks u smislu očekivanja jasnog i dosljednog rezultata u izbornom procesu.

## 6. Taktičko glasanje

Taktičko glasanje, poznato i kao strateško glasanje, događa se kada pojedinci biraju kandidata ili stranku koja nije njihova prva preferencija kako bi spriječili neželjeni ishod. Primjerice, u jednostavnim pluralnim izborima, glasači mogu podržati manje preferiranog, ali općenito popularnijeg kandidata kako bi postigli povoljniji rezultat. Ova politička strategija uključuje biranje preferencija na temelju taktičkih razloga umjesto vlastitih uvjerenja, često s ciljem podrške kandidatu s većim izgledima za pobjedu ili manje nepoželjnim osobinama u usporedbi s drugim opcijama (Wikipedia B, 2001).

Za primjeru, možemo uzeti situaciju biranja predstavnika grada gdje jedna skupina birača ima preferenciju  $B > A > C$  pri čemu je B njihov glavni odabir. Ali odlučuju promijeniti preferenciju u  $A > B > C$  jer su došli do zaključka da njihov kandidat ne može pobijediti ali može pobijediti kandidata C kojega ne žele za predstavnika. Stoga stavljaju kandidata A na prvu poziciju koji ima mogućnost da pobijedi.

## 7. Implementacija

```
def main():
    broj_grupa = int(input("Unesite broj skupina birača: "))
    preferencije = []

    for _ in range(broj_grupa):
        unos_preferencija = input("Unesite preferencije birača ove skupine (npr. ABCD): ")
        broj_glasaca = int(input("Unesite broj birača u ovoj skupini: "))
        preferencije.append([list(unos_preferencija), broj_glasaca])
```

Slika 2: Implementacija unos podataka (autorski rad)

Unos za svaku metodu je isti. Prvo se korisnika pita koliko skupina birača postoji te se u for petlji za taj svaku skupinu unose preferencije tako da se ako je preferencija određene skupine birača  $A > B > D > E > C$  u program unese ABDEC kao što je prikazuje *Slika 2*. Nakon toga unosi se broj birača koji imaju tu preferenciju te se obje varijable ***unos\_preferencija*** i ***broj\_glasaca*** dodaju u listu ***preferencije***

```
def bordina_metoda(preferencije):
    rezultati = Counter()
    for preferencija, broj_glasaca in preferencije:
        for i, kandidat in enumerate(preferencija):
            rezultati[kandidat] += broj_glasaca * (len(preferencija) - i - 1)
    return rezultati.most_common()
```

Slika 3: Implementacija Bordine metoda (autorski rad)

Funkcija ***bordina\_metoda*** kao argument također prima preferencije te se također koristi i Pythonova ugrađena biblioteka ***Counter*** koja služi za lakšu manipulaciju listama. Pomoću dvije for petlje se iterira kroz listu preferencija pri čemu druga for petlja u obzir uzima kandidata te njegov indeks s oznakom  $i$ . Rezultati kandidata za svaku preferenciju se zbrajaju pomoću formule za Bordinu metodu. Varijabla ***broj\_glasaca*** se množi s  $(len(preferencija) - i - 1)$  što znači da se od dužine preferencije koja predstavlja broj kandidata oduzima 1 i njegovo mjesto u toj listi te je tako implementiran Bordin algoritam. Na kraju se svi rezultati kandidata sortiraju silazno pomoću funkcije ***most\_common()***.

```
def vecinska_metoda(preferencije):
    rezultati = Counter()
    for preferencija, broj_glasaca in preferencije:
        rezultati[preferencija[0]] += broj_glasaca
    pobjednik = max(rezultati, key=rezultati.get)
    return pobjednik, rezultati[pobjednik]
```

Slika 4: Implementacija većinske metode (autorski rad)

Većinska metoda također kao argument prima preferencije te se za svaku preferenciju uzima prvi kandidat u listi te je na kraju pobjednik onaj kandidat kojeg je najviše glasača stavilo na prvo mjesto.

```

21 def condorcetov_pobjednik(preferencije):
22     broj_kandidata = len(preferencije[0][0])
23
24     pobjede = {}
25
26     for i in range(broj_kandidata):
27         for j in range(i + 1, broj_kandidata):
28             kandidat1 = chr(ord('A') + i)
29             kandidat2 = chr(ord('A') + j)
30
31             broj_glasaca1 = broj_glasaca2 = 0
32             for pref, broj_glasaca_preferencije in preferencije:
33                 index1 = pref.index(kandidat1)
34                 index2 = pref.index(kandidat2)
35
36                 if index1 < index2:
37                     broj_glasaca1 += broj_glasaca_preferencije
38                 elif index2 < index1:
39                     broj_glasaca2 += broj_glasaca_preferencije
40
41                 if broj_glasaca1 > broj_glasaca2:
42                     pobjede[kandidat1] = pobjede.get(kandidat1, 0) + 1
43                     G.add_edge(kandidat1, kandidat2, weight = broj_glasaca1)
44                 elif broj_glasaca2 > broj_glasaca1:
45                     pobjede[kandidat2] = pobjede.get(kandidat2, 0) + 1
46                     G.add_edge(kandidat2, kandidat1, weight = broj_glasaca2)
47
48     for kandidat in pobjede:
49         if pobjede[kandidat] == broj_kandidata - 1:
50             return kandidat
51
52     return None

```

Slika 5: Implementacija Condorcetove metode (autorski rad)

```

condorcetovPobjednik = condorcetov_pobjednik(preferencije)

if condorcetovPobjednik:
    print(f"Condorcetov pobjednik je: {condorcetovPobjednik}")
else:
    print("Ne postoji Condorcetov pobjednik.")

```

Slika 6: Pozivanje funkcije (autorski rad)

Nakon što je obavljen unos preferencija određenih skupina glasača pokreće se funkcija **condorcetov\_pobjednik** koja kao argument prima listu preferencije. Varijabla **broj\_kandidata** se dohvaća pomoću broja znakova u određenom retku liste pri čemu se prebrojavaju slova, primjerice ABCDE te se izračunava da postoji 5 kandidata. Potom se pomoću dvije for petlje uspoređuju svaka 2 kandidata, ali na način da se kandidati uspoređuju samo s onima koji dolaze nakon njih abecednim redom. Primjerice u prvoj iteraciji A će se usporediti sa svim ostalim kandidatima, dok će se B usporediti sa svima osim A kako se ne bi

ista usporedba vršila dva puta. C će se usporediti samo s D i E i tako dalje. Nakon toga pokreće se još jedna for petlja pomoću koje za svaku preferenciju pronalazimo indeks kandidata1 i kandidata2. Ako je indeks kandidata1 manji od indeksa kandidata2, to znači da je on u toj preferenciji bolje rangiran od strane glasača te mu se pripisuje broj glasova glasača te preferencije, točnije **broj\_glasaca\_preferencije**. Sukladno tome moramo dodati i pripadajući brid između vrhova na grafu. Na kraju kada se pregledaju sve skupine glasača (preferencije) kandidatu1 ili kandidatu2 će se pripisati pobjeda zavisno o tome tko je skupio veći broj glasova. Kada usporedimo sve kandidate onda možemo provjeriti je li određeni kandidati pobijedio protiv svih ostalih kandidata te ako je, onda je on Condorcetov pobjednik, a ako takav ne postoji onda ispisujemo prikladnu poruku.

```

82     kandidati = list(unos_preferencija)
83
84     G.add_nodes_from(kandidati)
85     pos = nx.shell_layout(G)
86
87     labels = nx.get_edge_attributes(G, 'weight')
88     nx.draw_networkx_nodes(G, pos, node_size=450, node_color='green')
89     nx.draw_networkx_labels(G, pos, font_size=12, font_color='black')
90     nx.draw_networkx_edges(G, pos, width=2, edge_color='black')
91     nx.draw_networkx_edge_labels(G, pos, edge_labels=labels)
92
93     plt.title("Preferencijalni graf kandidata")
94     plt.show()

```

Slika 7: Implementacija crtanja grafa (autorski rad)

Slika 7 prikazuje crtanje grafa pri čemu smo prvo kao kandidate označili listu iz unosa preferencija te smo ih dodali kao vrhove grafa pomoću funkcije **add\_nodes\_from(kandidati)**, što znači da će se unesena slova (kandidati) prikazati na grafu u obliku čvorova. Nakon toga smo oznake bridova dobili pomoću **get\_edge\_attributes** pri čemu kao oznaku na bridovima uzimamo **weight**, točnije broj glasova. Nakon toga pomoću **draw\_networkx\_nodes** i ostalih funkcija uređujemo graf prema vlastitim željama te na kraju dodajemo naslov grafa pomoću **plt.title** i prikazujemo ga na ekranu pomoću **plt.show()**.

## 7.1. Rješenje programa

Nakon što je implementacija završena, program se može koristiti za rješavanje zadataka. Za primjer će se uzet *Zadatak 3.2.1*. Slika 8 prikazuje unos početnih podataka, odnosno preferencija i broj birača po skupinama, dok Slika 9 prikazuje rješenja navedenog zadatka.

```

PS C:\Users\Korisnik\Desktop\python\dstg> python combine.py
Unesite broj skupina birača: 5
Unesite preferencije birača ove skupine (npr. ABCD): ABDEC
Unesite broj birača u ovoj skupini: 13
Unesite preferencije birača ove skupine (npr. ABCD): BADCE
Unesite broj birača u ovoj skupini: 7
Unesite preferencije birača ove skupine (npr. ABCD): CEBAD
Unesite broj birača u ovoj skupini: 5
Unesite preferencije birača ove skupine (npr. ABCD): DEABC
Unesite broj birača u ovoj skupini: 4
Unesite preferencije birača ove skupine (npr. ABCD): EACDB
Unesite broj birača u ovoj skupini: 1

```

Slika 8: Unos podataka u program (autorski rad)

```

Condorcetov pobjednik je: A

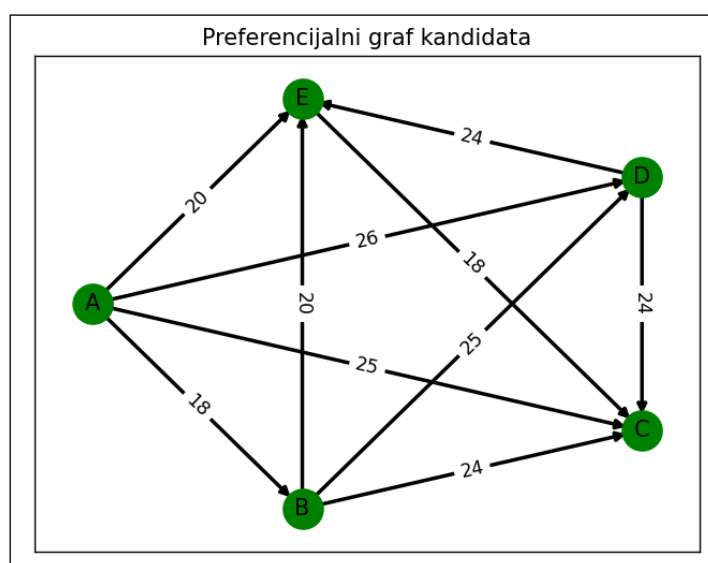
Borda rezultati:
[('A', 89), ('B', 81), ('D', 57), ('E', 44), ('C', 29)]

Većinska metoda rezultati:
('A', 13)

```

Slika 9: Rješenje zadatka (autorski rad)

Iz rješenja može se vidjeti da u ovom zadatku postoji Condorcetov pobjednik, a to je kandidat A. To znači da je kandidat A pobijedio u svim usporedbama s ostalim kandidatima. Kod Bordinog rezultata, također se može vidjeti da je A pobijedio s 89 bodova. U slučaju većinske metode također je kandidat A pobijedio. Ovdje je važno napomenuti da rezultat većinske metode može biti drugačiji od prethodne dvije jer uzima u obzir samo prve izbore, što je velika razlika u odnosu na Bordinu i Condorcetovu gdje se uzimaju u obzir svi podaci. U suštini, kandidat A je rješenje sve tri metode. Na kraju, program još ispiše preferencijalni graf kandidata koji je detaljnije opisan u *Poglavlju 2* ali se može vidjeti da je kandidat A Condorcetov pobjednik jer su mu svi bridovi izlazni.



Slika 10: Preferencijalni graf kandidata (autorski rad)

## 8. Arrowljev teorem mogućnosti

Arrowov teorem nemogućnosti sugerira da pod određenim uvjetima nije moguće doći do zajedničkog temelja uz razmatranje individualnih preferencija kroz funkciju socijalne skrbi. Prema Arrowovom teoremu nemogućnosti, u svim situacijama gdje su preferencije rangirane, nemoguće je konstruirati društveni poredak bez kršenja barem jednog od sljedećih uvjeta (Investopedia, 2023.):

1. Nediktatura: Nemoguće je formulirati društveni poredak koji uzima u obzir želje većine birača, sprječavajući istodobno pojedince ili male grupe da diktiraju konačne odluke.

2. Paretova učinkovitost: Ne može se postići društveni poredak koji uvijek poštuje jednoglasne individualne preferencije. Ako svaki birač preferira kandidata A nad kandidatom B, težnja je da kandidat A uvijek pobijedi, što može biti konfliktno s drugim uvjetima.

3. Neovisnost nebitnih alternativa: U slučaju uklanjanja jedne opcije, redoslijed preostalih alternativa ne bi trebao biti promijenjen. Ako je kandidat A rangiran ispred kandidata B, to bi trebalo ostati nepromijenjeno, čak i ako treći kandidat, kandidat C, bude eliminiran.

4. Neograničena domena: Glasovanje bi trebalo uzeti u obzir sve individualne preferencije, osiguravajući da su svi birači zastupljeni u formiranju društvenog poretka.

5. Društveni poredak: Svaki pojedinac bi trebao imati slobodu poredati izbore na bilo koji način i izraziti međusobne veze između različitih opcija, što može biti problematično kad se kombinira s ostalim uvjetima.

## Popis literature

ElectionBuddy, (bet dat). *Preferential Voting Systems* [Online]. Preuzeto 12.1.2024. s <https://electionbuddy.com/preferential/>

Investopedia (2023.) *Arrow's Impossibility Theorem Definition* [Online]. Preuzeto 12.1.2024. s <https://www.investopedia.com/terms/a/arrows-impossibility-theorem.asp>

OpaVote, (bez dat.). *Condorcet Voting* [Online]. Preuzeto 12.1.2024. s <https://www.opavote.com/methods/condorcet-voting>

The Editors of Encyclopaedia Britannica, (2008). *Plurality system* [Online]. Preuzeto 12.1.2024. s <https://www.britannica.com/topic/plurality-system>

Wikipedia A, (2001). *Borda Count* [Online] Preuzeto 12.1.2024. s [https://en.wikipedia.org/wiki/Borda\\_count](https://en.wikipedia.org/wiki/Borda_count)

Wikipedia B, (2001). *Strategic voting* [Online]. Preuzeto 12.1.2024. s [https://en.wikipedia.org/wiki/Strategic\\_voting](https://en.wikipedia.org/wiki/Strategic_voting)



## Popis slika

Slika 1: Preferencijalni graf kandidata (autorski rad) .....	6
Slika 2: Implementacija unos podataka (autorski rad) .....	11
Slika 3: Implementacija Bordine metoda (autorski rad) .....	11
Slika 4: Implementacija većinske metode (autorski rad) .....	11
Slika 5: Implementacija Condorcetove metode (autorski rad) .....	12
Slika 6: Pozivanje funkcije (autorski rad) .....	12
Slika 7: Implementacija crtanja grafa (autorski rad) .....	13
Slika 8: Unos podataka u program (autorski rad) .....	14
Slika 9: Rješenje zadatka (autorski rad) .....	14
Slika 10: Preferencijalni graf kandidata (autorski rad) .....	14

## Popis tablica

Tablica 1: Preferencije birača – Bordina metoda (autorski rad) .....	7
Tablica 2: Rješenje - Bordina metode (autorski rad) .....	8
Tablica 3: Preferencija birača – Condorcetova metoda (autorski rad) .....	8
Tablica 4: Rješenje - Condorcetova metoda (autorski rad) .....	9
Tablica 5: Preferencije birača – Većinska metoda (autorski rad) .....	9