

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
імені ТАРАСА ШЕВЧЕНКА**



ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра прикладних інформаційних систем

Звіт до лабораторної роботи №7

з курсу

«Системний аналіз та теорія прийняття рішень»

*студентки 3 курсу
групи ПП-33
спеціальності 122 «Комп'ютерні науки»
ОП «Прикладне програмування»
Матвіїв Анастасії Юріївни*

*Викладач:
Білий Р.О.*

Київ – 2023

Тема: методи голосування

Мета: вивчення методів голосування і дослідження їх властивостей; опанування методикою обробки профілів колективного голосування, знаходження переможця та відновлення колективного ранжування.

Кількість голосів	5	5	4	6
Впорядкування	A	C	B	C
	B	A	C	B
	C	B	D	D
	D	D	A	A

Правило (метод) відносної більшості. На перше місце 11 виборців поставили кандидата C ($n_C=11$), 5 виборців – A ($n_A=5$) і 4 виборці – B ($n_B=4$), ($n_D=0$).

Перемагає той кандидат, за якого проголосувала більшість виборців – C

Правило відносної більшості з вибуванням („відносна більшість у два тури”, „абсолютна більшість”). Якщо деякий кандидат набрав більше половини голосів ($50\%+1$), то він – переможець. Інакше у другий тур проходять два кандидати, що набрали відносну більшість голосів (тому – „відносна у два тури”). Для нашого профілю перемагає кандидат C (перемагає „абсолютно”, набираючи більше половини голосів)

Правило Борда („підрахунку очок”). У цьому правилі за останнє місце кандидата йому нараховується 0 балів (очок), за передостаннє – 1 бал, ..., за перше – $(m-1)$. Додаємо у профіль колонку з відповідними очками.

Кількість голосів	5	5	4	6	S
Впорядкування я	A	C	B	C	3
	B	A	C	B	2
	C	B	D	D	1
	D	D	A	A	0

Маємо: $nA = 5 \cdot 3 + 5 \cdot 2 = 15 + 10 = 25$

$nB = 5 \cdot 2 + 5 \cdot 1 + 4 \cdot 3 + 6 \cdot 2 = 10 + 5 + 12 + 12 = 39$

$nC = 5 \cdot 1 + 5 \cdot 3 + 4 \cdot 2 + 6 \cdot 3 = 5 + 15 + 8 + 18 = 46$

$nD = 4 \cdot 1 + 6 \cdot 1 = 4 + 6 = 10$

Перемагає кандидат, що набрав найбільшу кількість балів, у нашому випадку – це кандидат С.

Правило Кондорсе. За Кондорсе переможцем оголошується той кандидат, що „перемагає” всіх інших у парних порівняннях.

Маємо:

$a : b = 10:10,$

$b : c = 9:11,$

$c : d = 20:0,$

$a : c = 5:15,$

$b : d = 20:0 ,$

$a : d = 10:10.$

Єдиний кандидат, який „перемагає” всіх інших – це кандидат *c*.

Правило Копленда.

$$K(a)=K(a,b)+K(a,c)+K(a,d)=0-1+0=-1$$

$$K(b)=K(b,a)+K(b,c)+K(b,d)=0-1+1=0$$

$$K(c)=K(c,a)+K(c,b)+K(c,d)=+1+1+1=3$$

$$K(d)=K(d,a)+K(d,b)+K(d,c)=-1-1+0=-2$$

Отже, найвища оцінка у кандидата *c*, він і є переможцем за Коплендом.

Правило Сімпсона.

$$S(a)=\min(S(a,b), S(a,c), S(a,d))=\min(10,5,10)=5$$

$$S(b)=\min(S(b,a)+S(b,c)+S(b,d))=\min(10,9,20)=9$$

$$S(c)=\min(S(c,a)+S(c,b)+S(c,d))=\min(15,11,20)=11$$

$$S(d)=\min(S(d,a)+S(d,b)+S(d,c))=\min(10,0,0)=0$$

Найвища оцінка також у кандидата *C*, він і є переможцем за Сімпсоном.

Для даного профілю визначимо колективне голосування для усіх перелічених вище правил.

Для правила **відносної більшості** маємо, після виключення переможця *C* маємо профіль:

5	5	4	6
A	A	B	B
B	B	D	D

D	D	A	A
---	---	---	---

Для цього профілю переможцем буде кандидат А та В(нічия).

Для правила **Борда** маємо, після виключення переможця С маємо профіль:

5	5	4	6
A	A	B	B
B	B	D	D
D	D	A	A

Маємо: $nA = 5*2 + 5*2 = 10 + 10 = 20$

$nB = 5*1 + 5*1 + 4*2 + 6*2 = 5 + 5 + 8 + 12 = 30$

$nD = 4*1 + 6*1 = 4 + 6 = 10$

Для цього профілю переможцем буде кандидат В.

5	5	4	6
A	A	D	D
D	D	A	A

Для цього профілю нічия А та D.

Для правила **Кондорсе** маємо, після виключення переможця С маємо профіль:

5	5	4	6
A	A	B	B
B	B	D	D

D	D	A	A
---	---	---	---

$$a : b = 10 : 10,$$

$$b : d = 20 : 0 ,$$

$$a : d = 10 : 10.$$

Для цього профілю переможцем буде кандидат В.

5	5	4	6
A	A	D	D
D	D	A	A

Для цього профілю нічия А та D.

Для правила **Копленда** маємо, після виключення переможця С маємо профіль:

5	5	4	6
A	A	B	B
B	B	D	D
D	D	A	A

$$K(a) = K(a,b) + K(a,d) = 0 + 0 = 0$$

$$K(b) = K(b,a) + K(b,d) = 0 + 1 = 1$$

$$K(d) = K(d,a) + K(d,b) = -1 + 0 = -1$$

Для цього профілю переможцем буде кандидат В.

5	5	4	6
A	A	D	D

D	D	A	A
---	---	---	---

Для цього профілю нічия A та D.

$$K(a)=K(a,d)=0$$

$$K(d)=K(d,a)=0$$

Для правила **Сімпсона** маємо, після виключення переможця C маємо профіль:

5	5	4	6
A	A	B	B
B	B	D	D
D	D	A	A

$$S(a)=\min(S(a,b), S(a,d))=\min(10,10)=10$$

$$S(b)=\min(S(b,a)+S(b,d))=\min(10,20)=10$$

$$S(d)=\min(S(d,a)+S(d,b))=\min(10,0)=0$$

Для цього профілю нічия A та B.

Далі вирішила завдання за допомогою Python.

```
# Задані дані
votes = [
    ['A', 'B', 'C', 'D'],
    ['C', 'A', 'B', 'D'],
    ['B', 'C', 'D', 'A'],
    ['C', 'B', 'D', 'A']
]

amount_of_votes_sequence = [5, 5, 4, 6]

# Кількість виборців та кандидатів
n = len(amount_of_votes_sequence)
m = len(votes)

def print_scores(scores):
```

```

    for candidate, score in scores.items():
        print(f"{candidate}: {score}")

scores_majority = {candidate: 0 for candidate in votes[0]}

# Підрахунок кількості голосів за кожного кандидата на першому місці
for i in range(n):
    candidate = votes[i][0]
    scores_majority[candidate] += amount_of_votes_sequence[i]

winner_majority = max(scores_majority, key=scores_majority.get)

print_scores(scores_majority)
print(f"Переможець: Кандидат {winner_majority} з
{scores_majority[winner_majority]} голосами")
scores_absolute_majority = {candidate: 0 for candidate in votes[0]}
# Підрахунок кількості голосів за кожного кандидата на першому місці
for i in range(n):
    candidate = votes[i][0]
    scores_absolute_majority[candidate] += amount_of_votes_sequence[i]

# Знаходження кандидата з більшістю голосів
majority_threshold = n // 2 + 1
winner_absolute_majority = max(scores_absolute_majority,
key=scores_absolute_majority.get)

# Перевірка на абсолютну більшість
if scores_absolute_majority[winner_absolute_majority] >= majority_threshold:
    print(f"Переможець: Кандидат {winner_absolute_majority} з
{scores_absolute_majority[winner_absolute_majority]} голосами (абсолютна
більшість)")
else:
    # Знаходження двох кандидатів з відносною більшістю для другого туру
    second_round_candidates = sorted(scores_absolute_majority,
key=scores_absolute_majority.get, reverse=True)[:2]
    print(f"Другий тур: Кандидати {second_round_candidates}")
    scores_borda = {candidate: 0 for candidate in votes[0]}

borda_weights = [3, 2, 1, 0]

# Розрахунок балів за методом Борда
for i in range(len(votes)):
    for j in range(len(votes[i])):
        scores_borda[votes[i][j]] += borda_weights[j] *
amount_of_votes_sequence[i]

print_scores(scores_borda)

winner_borda = max(scores_borda, key=scores_borda.get)

```



```

print(f"Переможець: Кандидат {winner_borda} з {scores_borda[winner_borda]} балами за методом Борда")
# Функція для порівняння двох кандидатів
def compare_candidates(candidate1, candidate2):
    count_winner1 = 0
    count_winner2 = 0

    for vote in votes:
        if vote.index(candidate1) < vote.index(candidate2):
            count_winner1 += amount_of_votes_sequence[votes.index(vote)]
        if vote.index(candidate1) > vote.index(candidate2):
            count_winner2 += amount_of_votes_sequence[votes.index(vote)]

    return count_winner1, count_winner2

# Ініціалізація кількості перемог для кожного кандидата
wins = {candidate: 0 for candidate in votes[0]}

# Порівняння кандидатів і підрахунок перемог
candidates = list(wins.keys())
for i in range(len(candidates)):
    for j in range(i + 1, len(candidates)):
        wins_candidate1, wins_candidate2 = compare_candidates(candidates[i],
candidates[j])
        if wins_candidate1 > wins_candidate2:
            wins[candidates[i]] += 1
        if wins_candidate1 < wins_candidate2:
            wins[candidates[j]] += 1

winner = max(wins, key=wins.get)

print(f"Переможець за методом Кондорсе: {winner}")
print_scores(wins)
# Функція для порівняння двох кандидатів
def compare_candidates_kop(candidate1, candidate2):
    count_1 = 0
    count_2 = 0

    for vote in votes:
        if vote.index(candidate1) < vote.index(candidate2):
            count_1 += amount_of_votes_sequence[votes.index(vote)]
        elif vote.index(candidate1) > vote.index(candidate2):
            count_2 += amount_of_votes_sequence[votes.index(vote)]

    if count_1 == count_2:
        return 0
    if count_1 > count_2:
        return 1
    if count_1 < count_2:

```

```

        return -1

# Ініціалізація оцінок Копленда для кожного кандидата
copeland_scores = {candidate: 0 for candidate in votes[0]}

# Порівняння кандидатів і підрахунок оцінок Копленда
candidates = list(copeland_scores.keys())
for i in range(len(candidates)):
    for j in range(len(candidates)):
        if i != j:
            score = compare_candidates_kop(candidates[j], candidates[i])
            copeland_scores[candidates[j]] += score

winners = [candidate for candidate, score in copeland_scores.items() if score ==
max(copeland_scores.values())]

print(f"Переможець Копленда: {'', '.join(winners)}")
print_scores(copeland_scores)

# Функція для порівняння двох кандидатів
def compare_candidates_sim(candidate1, candidate2):
    count_winner1 = 0
    count_winner2 = 0

    for vote in votes:
        if vote.index(candidate1) < vote.index(candidate2):
            count_winner1 += amount_of_votes_sequence[votes.index(vote)]
        if vote.index(candidate1) > vote.index(candidate2):
            count_winner2 += amount_of_votes_sequence[votes.index(vote)]

    return count_winner1

# Ініціалізація оцінок Сімпсона для кожного кандидата
simpson_scores = {candidate: 0 for candidate in votes[0]}

# Порівняння кандидатів і підрахунок оцінок Сімпсона
candidates = list(simpson_scores.keys())

for i in range(len(candidates)):
    result_of_sim = list()
    for j in range(len(candidates)):
        if i != j:
            score = compare_candidates_sim(candidates[i], candidates[j])
            result_of_sim.append(score)
    simpson_scores[candidates[i]] = min(result_of_sim)

winners = [candidate for candidate, score in copeland_scores.items() if score ==
max(copeland_scores.values())]

```

```
print(f"Переможець Сімпсона: {' '.join(winners)}")
```

Висновок:

Під час виконання роботи я вивчила методи голосування і дослідження їх властивостей; опанувала методикою обробки профілів колективного голосування, знайшла переможця та відновила колективне ранжування.

Відповіді на контрольні питання:

1. Аксіоми для "розумних" "логічних" вимог до правил голосування:

Повнота (Completeness): Усі альтернативи повинні бути порівняні між собою.

Транзитивність (Transitivity): Якщо вибір А переважає В, і В переважає С, то А також повинно переважати С.

Рефлексивність (Reflexivity): Вибір А повинен переважати сам себе.

2. Аксіоми для властивостей голосування:

Незалежність альтернатив (Independence of Irrelevant Alternatives): Результат голосування не повинен змінитися внаслідок видалення або додавання альтернатив, які не отримали перемогу.

Слабка групова раціональність (Weak Group Rationality): Якщо більшість вибирає А перед В, то колективне рішення також повинно вибирати А перед В.

3. Правило голосування та критерії:

Утилітаризм: Відповідає принципу максимізації загальної користі. Мета полягає в тому, щоб обрана альтернатива максимізувала загальну користь для всіх учасників.

Егалітаризм функції колективної корисності: Заснований на ідеї зменшення різниці в благополуччі між учасниками голосування.

4. Парадокси голосування та маніпуляції:

Парадокс вибору (Arrow's Impossibility Theorem): Показує, що неможливо задовольнити всі вищезазначені аксіоми одночасно, існуючи при цьому логічною груповою процедурою визначення вибору.

Маніпуляції: Учасники можуть стратегічно висловлювати свої уподобання з метою впливу на результат голосування.

5. Правила голосування в різних ситуаціях:

Спортивні змагання: Зазвичай використовують правила більшості, де перемагає та альтернатива, яку підтримує більшість.

Вибір депутатів: Використовуються різні системи, такі як мажоритарна або пропорційна системи в залежності від країни та контексту.

Вибір президента: Може використовуватися система голосування з використанням виборчих колегій або прямого голосування.