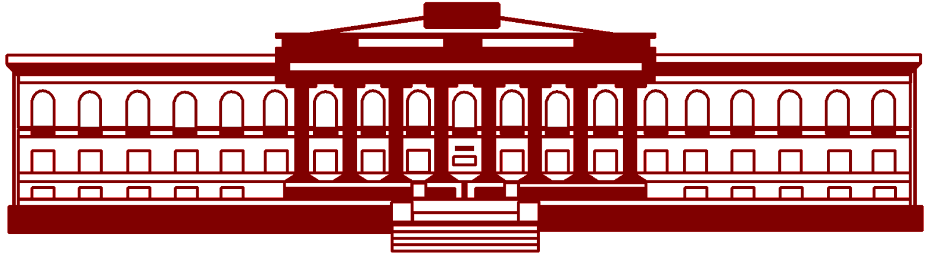
**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**

**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-31*

*спеціальності 122 «Комп'ютерні науки»*

*ОП «Прикладне програмування»*

Голотюк Вікторії Миколаївни

*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 (nC*=11) , 5 виборців – *A(nA*=5) і 4 виборці – *B(nB*=4), (*nD*=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). Добавимо у профіль колонку з відповідними очками.

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

Маємо: nA= 5\*3 + 5\*2 = 15 + 10 = 25

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

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

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

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

**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 с, він і є переможцем за Комплендом.

**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 A та B(нічия).

Для правила **Борда** маємо, після виключення переможця *C* 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 B.

|  |  |  |  |
| --- | --- | --- | --- |
| 5 | 5 | 4 | 6 |
| A | A | D | D |
| D | D | A | A |

Для цього профілю нічия A та D.

Для правила **Кондорсе** маємо, після виключення переможця *C* 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 B.

|  |  |  |  |
| --- | --- | --- | --- |
| 5 | 5 | 4 | 6 |
| A | A | D | D |
| D | D | A | A |

Для цього профілю нічия A та D.

Для правила **Копленда** маємо, після виключення переможця *C* 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 B.

|  |  |  |  |
| --- | --- | --- | --- |
| 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 A переважає B, і B переважає C, то A також повинно переважати C.

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

1. 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 A перед B, то колективне рішення також повинно вибирати A перед B.

1. 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.

1. 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.

1. 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.