Міністерство освіти і науки України

Національний технічний університет України

«Київський політехнічний інститут»

Інститут Прикладного системного аналізу

Кафедра Системного проектування

Лабораторна робота №4

з дисципліни «Теорія прийняття рішень»

«Прийняття рішень за допомогою методів голосування»

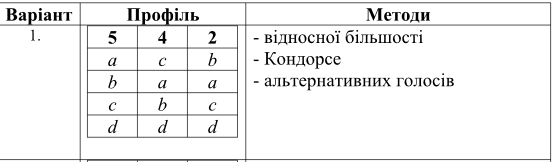
Виконала:   
студентка групи ДА-42 Балан Катерина

Київ - 2017

# Мета роботи

# Ознайомитись з методами прийняття рішень за допомогою методів голосування

# Варіант №1



**Аналітичний рахунок**

Відносної більшості a(n = 5) , c(n = 4), b(n = 2), d( n = 0). Перемагає кандидат а.

Кондорсе a:b(9:2) a:c(7:4) a:d(11:0) – Перемагає клієнт а.

Альтернативних голосів – найгірший кандидат D(11) виключаємо його із таблиці

|  |  |  |
| --- | --- | --- |
| **5** | **4** | **2** |
| A | C | B |
| B | A | A |
| C | B | C |
| D | D | D |

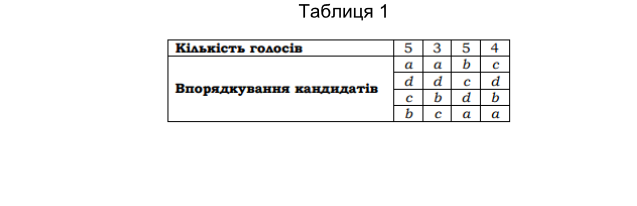
Найгірший кандидат С(7) виключаємо його із таблиці

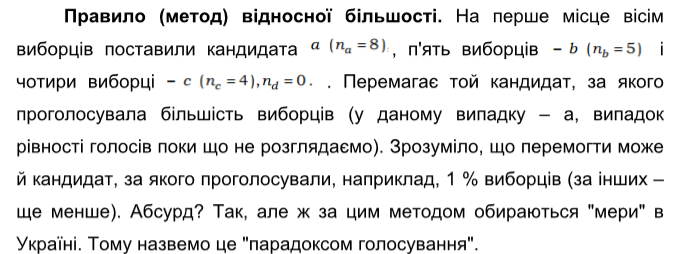
|  |  |  |
| --- | --- | --- |
| **5** | **4** | **2** |
| A | C | B |
| B | A | A |
| C | B | C |

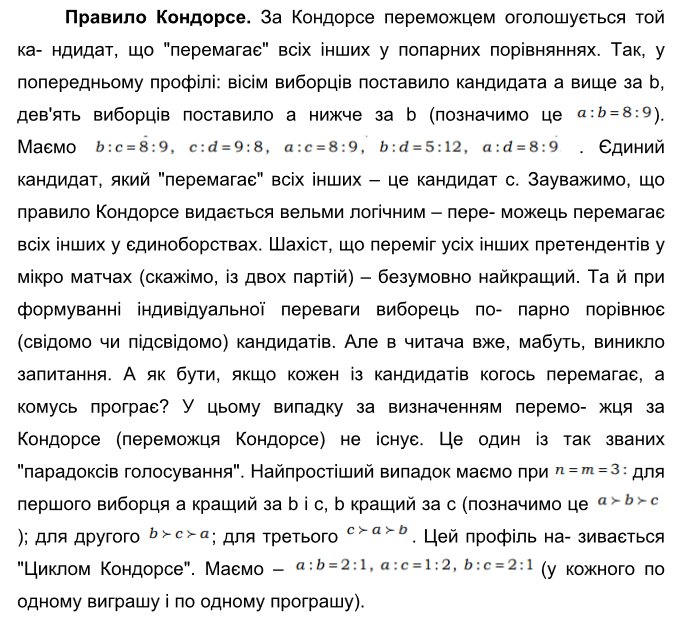
Найгірший кандидат B(9) виключаємо його із таблиці – Перемагає кандидат А

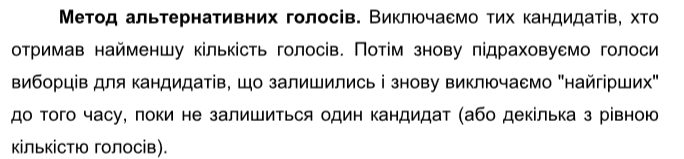
|  |  |  |
| --- | --- | --- |
| **5** | **4** | **2** |
| A | A | B |
| B | B | A |

# Короткі теоретичні відомості









# Виконання роботи

Результати розрахунку

/usr/bin/python2.7 /home/katya/workspace/Decision\_theory/lab\_tpr4/lab4.py

Profiles:

['a', 'c', 'b']

['b', 'a', 'a']

['c', 'b', 'c']

['d', 'd', 'd']

Profiles weight: [5, 4, 2]

Winner (relative majority): a

Winner (Condorcet): a

Winner (method of alternative votes): a

Process finished with exit code 0

**Лістинг програми**

[**https://github.com/katebalan/Decision\_theory/tree/master/lab\_tpr4**](https://github.com/katebalan/Decision_theory/tree/master/lab_tpr4)

**Висновки** : За правилом відносної більшості перемагає кандидат за якого проголосувала більшість виборців, але можливий так званий «парадокс голосування» коли перемагають кандидати з менш чи 1% більшості. За правилам Кондорсе перемагає кандидат який переміг всіх в попарних порівняннях, проте можливий випадок коли переможця за даним правилом знайти неможливо. За методом альтернативних голосів на кожному етапі ми виключаємо найгірших і залишається лиш найкращий кандидат, проте можливий варіант коли буде декілька переможців за даним методом.

**Лістинг програми**

import itertools

from collections import defaultdict

document1 = open('1.txt', 'r')

profiles = []

print "Profiles: "

for line in document1:

profiles.append([])

for symbol in line.split():

profiles[len(profiles) - 1].append(symbol)

print profiles[len(profiles) - 1]

document1.close()

document2 = open('2.txt', 'r')

profiles\_weight = []

for number in document2.readline().split():

profiles\_weight.append(int(number))

document2.close()

print "Profiles weight: {}".format(profiles\_weight)

def relative\_majority(profiles, profiles\_weight):

result = defaultdict(int)

for i in range(len(profiles[0])):

result[profiles[0][i]] += profiles\_weight[i]

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

return winner

def condorcet(profiles, profiles\_weight):

profiles\_tranc = zip(\* profiles)

results = defaultdict(int)

for iter in range(len(profiles\_weight)):

for c1, c2 in itertools.combinations(profiles\_tranc[iter], 2): # combinations('ABCD', 2) --> AB AC AD BC BD CD

assert (c1 != c2)

mini, maxi, res = (c1, c2, profiles\_weight[iter]) if c1 < c2 else (c2, c1, -profiles\_weight[iter])

results[mini, maxi] += res

for c in {c for profile in profiles\_tranc for c in profile}:

# TO DO understand if statement

if (all((res > 0 if c == mini else res < 0) for (mini, maxi), res in results.items() if c in [mini, maxi])):

return c

return "can't be find"

def alternative\_votes\_method(profiles, profiles\_weight):

result = defaultdict(int)

profiles\_count = len(profiles[0])

last = len(profiles[0]) - 1

while( last > 1 ):

for iter in range(profiles\_count):

result[profiles[last][iter]] += profiles\_weight[iter]

worst = max(result, key=result.get)

profiles\_tranc = [list(i) for i in zip(\* profiles)]

for profile in profiles\_tranc:

profile.remove(worst)

profiles = [list(i) for i in zip(\* profiles\_tranc)]

result.clear()

last -= 1

return relative\_majority(profiles, profiles\_weight)

print "Winner (relative majority): {}".format(relative\_majority(profiles, profiles\_weight))

print "Winner (Condorcet): {}".format(condorcet(profiles, profiles\_weight))

print "Winner (method of alternative votes): {}".format(alternative\_votes\_method(profiles, profiles\_weight))