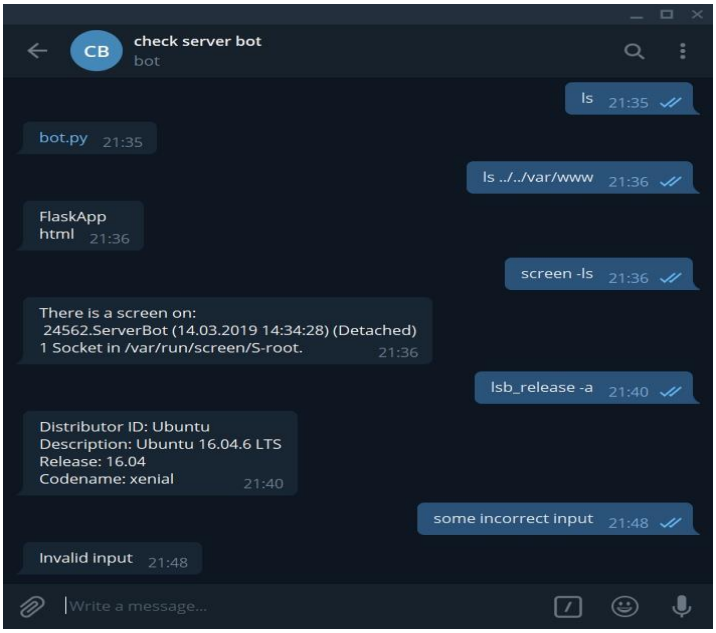

Рис 1. Приклад роботи бота

Список літературних джерел

1. webrtc.org [Електронний ресурс] – Режим доступу до ресурсу: https://webrtc.org/
2. NAT [Електронний ресурс] – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Network_address_translation
3. pyTelegramBotAPI [Електронний ресурс] – Режим доступу до ресурсу: https://github.com/eternnoir/pyTelegramBotAPI
4. Python docs [Електронний ресурс] – Режим доступу до ресурсу: https://www.python.org/doc
5. xterm [Електронний ресурс] – Режим доступу до ресурсу: https://www.x.org/releases/X11R6.7.0/doc/xterm.1.html
6. os.forkpty() [Електронний ресурс] – Режим доступу до ресурсу: https://www.programcreek.com/python/example/10221/os.forkpty

УДК 004.4(043.2)

Стиценко Н.Д., студентка 2 курсу
СО «Магістр» спеціальності
113 «Прикладна математика»
Скрипник А.В., студент 2 курсу
СО «Магістр» спеціальності
113 «Прикладна математика»
Вєтров О.С., старший викладач кафедри прикладної математики

ЗАСТОСУВАННЯ ДЕЯКИХ ПРИКЛАДНИХ КОМБІНАТОРНИХ АЛГОРИТМІВ В ЗАДАЧАХ КОЛЕКТИВНОЇ УХВАЛИ РІШЕННЯ

Донецький національний університет імені Василя Стуса, м. Вінниця

Комбінаторика, як відомо [1], це наука про перерахунок та перелічення елементів у скінченних множинах. Класичні теоретичні задачі комбінаторики стосуються, як правило, проблеми перерахунку на скінчені множині елементів, що мають деякий набір заданих властивостей (такі елементи ще називають конфігураціями). Для прикладних задач найбільш актуальною є задача не стільки перерахунку об'єктів (в простих випадках це можна дуже просто зробити по відомих формулах чи їх нескладних композиціях), а більше перелічення елементів із заданими властивостями. Тобто програміста цікавить перш за все предмет перелічувальної комбінаторики.

Комбінаторні алгоритми [2] знаходять широкий застосунок у прикладних задачах, коли необхідно опрацювати різноманітні конфігурації об'єктів, заданих на певній множині. Конфігурації формуються програмно за допомогою певного алгоритму, що задає користувач. Саме розробник алгоритму має визначити, яким способом генерується n+1 конфігурація, виходячи з передніх. Критичною в цьому випадку є обчислювальна ефективність алгоритму: його часова складність [3]. Найчастіше комбінаторні алгоритми мають експоненціальну складність (клас NP), і вже на відносно невеликих обсягах вхідних даних можуть становити задачу, що фактично неможливо реалізувати на сучасному рівні розвитку обчислювальної техніки.

В якості прикладу для дослідження була обрана задача порівняння електоральних конфігурацій для ухвалення колективних рішень по методу Кондерсе та методу Борда. Узагальнимо згадані правила як це описано в [4]. Задається неспадаюча послідовність дійсних чисел $s_0 \le s_1 \le ... \le s_{m-1}$, при цьому строго $s_0 < s_{m-1}$.. Виборці ранжують кандидатів згідно із власними уподобаннями, причому s_0 балів дається за останнє місце, s_1 – відповідно за передостаннє і т. д. У підсумку, обирається кандидат із максимальною загальною сумою балів. Описана процедура є "узагальненим правилом Борда".

Порівнюються правила Кондорсе й Борда. Їх можна вважати несумісним, оскільки існують профілі, при яких переможець по методу Кондорсе не може бути переможцем Борда ні за якою системою розподілу балів.

Розглянемо профіль [4]. Вважаємо $s_0 < s_1 < s_2$.

Голоси/Бали	2	1	1	3
s_2	A	A	B	C
s_1	B	C	C	A
s_0	C	B	A	B

m(A): $3p_2 + 3p_1 + p_0$
m(B): $p_2 + 2p_1 + 4p_0$
m(C): $3p_2 + 2p_1 + 2p_0$

Для цього профілю переможець методом Кондорсе кандидат С. За методом Борда бачимо що при строгих нерівностях $s_0 < s_1 < s_2$ переможцем є очевидно кандидат А.

Твердження залишається справедливим і у випадку $s_0 \le s_1 \le s_2$, $s_0 < s_2$. У [4] запропонований профіль, що задовольняє заданій вище умові.

Голоси/Бали	4	2	3	2
s_2	A	B	B	C
s_1	B	A	C	A
s_0	C	C	A	B

m(A): $4s_3 + 4s_1 + 3s_0$
m(B): $5s_3 + 4s_1 + 2s_0$
m(C): $2s_3 + 3s_1 + 6s_0$

Профіль описує ситуацію з 11 виборцями та 3 кандидатами, тоді як мінімальний до цього профіль [5] містив 17 виборців і три кандидати.

Була поставлена задача визначити, чи наведений в [4] профіль є оптимальним з точки зору мінімально можливої виборців, чи є оптимальний профіль єдиним.

Для розв'язку поставленої задачі були використані можливості мови Python, зокрема модуль itertools, що містить оптимізовані програмні реалізації базових комбінаторних алгоритмів. Додатково був реалізований алгоритм розбиття натурального числа на доданки [2], для генерації усіх можливих конфігурацій розподілу виборців за 6 варіантами профілю.

Список літературних джерел.

1. Бардачов Ю.М., Соколова Н.А., Ходаков В.Є. Дискретна математика. – К.: Вища школа, 2007. – 383 с.
2. Кнут Д.Э. Искусство программирования. Том 4, А. Комбинаторные алгоритмы. Часть 1. – М.: Вильямс, 2013. – 960 стр.
3. Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К. Алгоритмы. Построение и анализ. – М: Вильямс, 2013 год – 1324 стр.
4. Волошин О.Ф., Мащенко С.О. Моделі та методи прийняття рішень. – К.: Видавничо-поліграфічний центр "Київський університет", 2010. – 336 с.
5. Мулен Э. Кооперативное принятие решений: Аксиомы и модели. – М., 1991. – 464 стр.

УДК 004.4(043.2)

Мартьянова Т.А., старший викладач, к.т.н., кафедра комп'ютерних наук та інформаційних технологій
Загоруйко Л.В., доцент, к.т.н, кафедра комп'ютерних наук та інформаційних ехнологій
Човган Д.С., студент 4 курсу спеціальності 122 «Комп'ютерні науки»

ЦЕНТРАЛЬНИЙ МОДУЛЬ КЕРУВАННЯ МІСЬКОЮ СИСТЕМОЮ ШЕРІНГУ ПАРАСОЛЬОК

Донецький національний університет імені Василя Стуса, м. Вінниця

Метою даної роботи є – система шерінгу парасоль. А саме її центральний модуль керування.

Важливу роль у прийняті рішення про створення системи шерінгу відіграла статистика їх використання. Адже саме шерінг є виходом із великої кількості ситуацій до якої людина буде не готова.