КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА ШЕВЧЕНКА
Факультет комп'ютерних наук та кібернетики
Кафедра Дослідження операцій

Знаходження оптимального шляху обміну у біржах на основі маркет-мейкерів функціями обміну константного добутку

студента 4-го курсу
Байбули Кирила Аленовича

Київ — 2024

РЕФЕРАТ

БІРЖІ, МАРКЕТ-МЕЙКЕРИ, AMM, КОНСТАНТНІ ФУНКЦІЇ ОБМІНУ, ДИНАМІЧНІ ГРАФИ, ОПТИМІЗАЦІЯ, АЛГОРИТМИ, ПРОТОКОЛИ.

Дана робота присвячена дослідженню методів обробки бірж основаних на методі автоматизованих маркет-мейкерів (з англ. “Automated Market Maker”) AMM і їх часткового випадку із константними функціями обміну. Головною метою роботи є розробка та імплементація алгоритму знаходження найоптимальнішого шляху для динамічного графа подібної біржі. Оптимальним буде вважатися шлях що між всіма можливими шляхами пари валют X/Y буде отримувати найбільшу кількість Y за X.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ

Позначення	Значення
ААМ	Біржі основані на методі автоматичних маркет-майкерів.
ММКФ	Маркет мейкери із константними функціями
X/Y	Пара на біржі валюти типу X та Y
$X \implies Y$	Короткий запис обміну валюти X на Y

1. ВСТУП

У світі фінансів і торгівлі цінними паперами, централізовані біржі завжди відігравали ключову роль, забезпечуючи місцезнаходження та централізовану інфраструктуру для трейдерів та інвесторів. Проте останнім часом відбувається зростання інтересу до децентралізованих бірж, що призводить до переворотних змін у фінансовому секторі. Децентралізовані біржі стають ключовим елементом цієї нової економічної парадигми. Підкріпленні математикою та детермінованими правилами, децентралізовані біржі забезпечують високу швидкість та надійність для їх користувачів та завдяки відкритості дають нові можливості для нових незалежних гравців ринку.

У данній роботі ми розглянемо біржі основані на методі автоматизованих маркет-мейкерів (з англ. “Automated Market Maker”) найдавніша згадка котрих датується ще 1956 [7]. Конкретно розглянемо простий для аналізу і один з найпопулярніших по кількості імплементацій варіант на константних функціях [3].

На відміну від традиційних бірж, де ціна визначається за допомогою зіставлення заявок купівлі та продажу, AAM використовують алгоритм для визначення відношення вартості валют через кількості вкладів у валютну пару. Це дозволяє створити вигідну систему для владників ліквідності, що отримують винагороду за свої вклади, а також для бажаючих скористатися ліквідністю для обміну.

Також основною цілю нашого дослідженню буде біржа Uniswap V2 [1], котра є однією з найпопуляніших імплементацій AMM на даний момент.

1.1. Загальна математична модель

1.1.1. Константна функція обміну

ММКФ відносно просто описати аналітично, тому багато дослідників і практиків працюють з ними саме в такому “аналітичному” середовищі. Модель складається з двох основних об’єктів: торгової функції $f : \mathbf{R}_+^n \mapsto \mathbf{R}$ та вектора запасів $R \in \mathbf{R}_+^n$. Користувач пропонує обмін, представлену у вигляді портфеля $\Delta \in \mathbf{R}_n$, і обмін вважається дійсним, якщо торгова функція, оцінена на резервах після завершення обміну, має те саме значення, що і функція, оцінена на резервах до завершення угоди, тобто, якщо $f(R - \Delta) = f(R)$. (Звідси і назва “маркет-мейкер з постійною функцією”.) Якщо ця рівність виконується, то ММКФ виплачує користувачеві Δ, що призводить до появи нових резервів $R - \Delta$. (Якщо угода є недійсною, користувач нічого не отримує і не виплачує.) Постачальники ліквідності, які надають резерви R, під які здійснюються угоди, заробляють на цих угодах комісійні. Той факт, що цей процес легко описати і реалізувати, а також те, що він має багато сильних теоретичних гарантій, став однією з причин його успіху, особливо в таких складних для захисту середовищах, як публічні блокчейни. Незважаючи на простий опис, ММКФ породили велику

кількість досліджень їх фінансових, арбітражних і маршрутних властивостей (наприклад, [2], [4], серед багатьох інших).

У нашому випадку розглядається імлементація ММКФ від Uniswap:

$$R_x \cdot R_y = k = const \tag{1}$$

де R_x кількість валюти (резерви від англ. "reserves") X, та R_y кількість валюти Y у парі X/Y, а k — деяка константа котра задається при створенні пари на біржі. Що є маркет мейкером константного добутку [11] (з англ. *"CPMM Constant Product Market Maker"*), де відношення вкладів до і після має бути константним. Тобто при продажі Δx валюти отримуємо Δy:

$$R_x \cdot R_y = (R_x + \Delta x) \cdot (R_y - \Delta y)$$

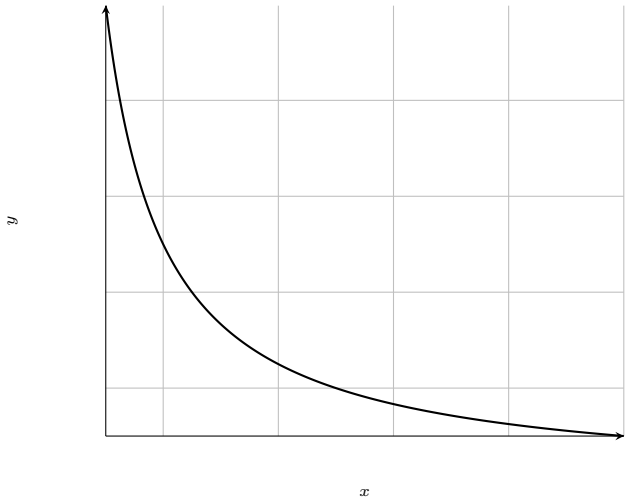

Рис. 1: *Зображення графіку залежностей вкладів у пару*

1.1.2. Біржевий граф

Нехай $G_t = (V_t, E_t)$ — неорієнтований динамічний граф, де V_t — множина вершин таких, що кожна вершина є конкретною валютою на біржі, E_t — множина ребер цього графа, тобто кожне ребро відображує те, чи є можливість на біржі обміняти деякі дві різні валюти з V_t, у момент часу t.

Так як в залежності від часу на біржі можуть з'являтися та зникати як валюти так і зв'язки (шляхи обміну) між ними, то ми описуємо біржу у вигляді динамічного графа, що залежить від часу t [8]. Вага ребра визначається я функція від вкладів у пару на момент часу t та об'єму обміну при проходженні через дане ребро (про що мова йде далі).

У біржі існує два типи подій, що змінюють стан біржі:

1. Створення нової пари (змінює ребра та вершини).
2. Обмін однієї пари або послідовності валют на біржі (змінює ваги ребер).

2. Функція обміну

У даному розділі ми розширемо та формалізуємо модель описану у вступі (1.1).

2.1. Об'єм обміну при одному обміні в парі

Лема 1. *Нехай існує на біржі пара X/Y із об'ємом R_X вкладів валюти X та R_Y вкладів валюти Y. Тоді об'єм y валюти Y при вкладанні x валюти X у пару X/Y дорівнює:*

$$y = \frac{R_Y x}{(R_X + x)} \tag{2}$$

Доведення. Розглянувши (1) неважко вивести формулу отримуємої кількості y валюти Y при вкладанні x кількості валюти X у пару X/Y із вкладами R_X, R_Y. Звідси з (1) до обміну відношення було:

$$R_X \cdot R_Y = k$$

Після вкладу нової кількості x валюти в пару, отримуємо невідому кількість y з вкладу об'ємом R_X. При цьому по правилам протоколу відношення має залишатися незмінимим до і після обміну, тобто дорівнювати тому ж самому k, отже:

$$(R_X + x) \cdot (R_Y - y) = k$$

Прирівнявши обидві рівності, отримаємо:

$$\begin{aligned} (R_X + x) \cdot (R_Y - y) &= R_X \cdot R_Y \\ (R_X - y) &= \frac{R_X R_Y}{(a + x)} \\ (R_Y - y) &= \frac{R_X R_Y}{(R_X + x)} \\ y &= R_Y - \frac{R_X R_Y}{(R_X + x)} \end{aligned} \tag{3}$$

Або з (3) більш компактний варіант:

$$\begin{aligned} y &= \frac{R_Y R_X + R_Y x - R_X R_Y}{(R_X + x)} \\ y &= \frac{R_Y x}{(R_X + x)} \end{aligned} \tag{4}$$

■

Тепер розглянемо випадок коли існує деяке $0 \leq \rho < 0$ що визначає комісію пари (наприклад для UniswapV2 це значення є константним 0.003 або 0,3%). Тоді формула обміну буде:

$$y = \frac{R_Y x\gamma}{R_X + x\gamma}$$

де $\gamma = 1 - \rho$ або:

$$y = R_Y - \frac{R_Y R_X}{R_X + x\gamma} \tag{5}$$

2.2. Композиція функцій обміну

Перед наступним розділом пропонуємо розглянути властивості та сутність композицій функцій обміну. Нехай необхідно знайти об'єм обміну $X \implies Y$, з (5) це буде:

$$y = f_{X/Y}(x) = R_Y - \frac{R_Y R_X}{R_X + x\gamma}$$

Обмін $Y \implies Z$:

$$z = f_{Y/Z}(y) = R_Z - \frac{R_Z R_Y}{R_Y + y\gamma}$$

Зафіксуємо об'єм Δx валюти X для отримання Δy валюти Y:

$$\Delta y = f_{X/Y}(\Delta x) \tag{6}$$

І для наступного Δy на Δz:

$$\Delta z = f_{Y/Z}(\Delta y)$$

З (6):

$$\Delta z = f_{Y/Z}(\Delta y) = f_{Y/Z}(f_{X/Y}(\Delta x)) = (f_{X/Y} \circ f_{Y/Z})(\Delta x)$$

Звідси робимо висновок що композиція функція обміну, є аналогічним до обміну по парам цих функцій.

2.3. Об'єм отримуємий при n-тій кількості переходів.

У кінцевій задачі ми будемо розглядати переходи між n-тою кількістю валют:

Лема 2. *Нехай, R_{C_i} — об'єм вкладів пари **на котру** ми вносимо валюту C_i, а $R_{C_{i+1}}$ — об'єм вкладів пари **з котрої** ми виносимо валюту C_{i+1} при i-тому обміні, де $i = \overline{0, n}$. Тоді об'єм обміну $C_0 \implies C_1 \ldots \implies C_{n-1} \implies C_n$ при вхідному x:*

$$y = (f_{C_0/C_1} \circ f_{C_1/C_2} \circ \ldots f_{C_i/C_{i+1}})(x) = \\ = x \prod_{i=1}^{n} R_{C_{i+1}} \div \left(\prod_{i=1}^{n} R_{C_i} + x \sum_{i=0}^{n-1} \left(\prod_{k=1}^{i} R_{C_{i+1}} \cdot \prod_{j=i+2}^{n} R_{C_i} \right) \right) \tag{7}$$

Результат дозволяє в один математичний вираз описати декілька переходів, проте при обрахуванні методами програмного забезпечення добутки вкладів можуть бути настільки великими числами (навіть два обміни між парами із вкладами по 10^6 утворює числа $(10^6)^3$), що при використанні чисел із обмеженою точністю можуть утворювати переповнення.

На момент написання цієї роботи, максимальний розмір регістра середньостатистичного комерційного комп'ютера скаладав 64 біта, де максимальне значення числа без знаку становить $2^{64} < 10^{20}$.

2.4. Графічний зміст

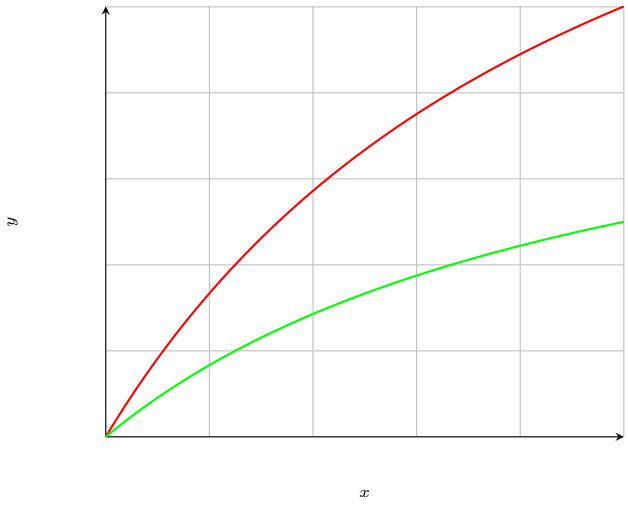

Рис. 2: *Зображення графіку залежностей вкладів у пару*

3. Формалізація графа

Так як головною метою цієї роботи є опис загального алгоритму для знаходження найоптимальнішого шляху для обміну, ми спробуємо формалізувати визначення ваги у графі $G_t = (V_t, E_t)$.

Розглянувши формулу обміну (2) ми можемо побачити гіперболічну залежність вкладів однієї валюти від іншої.

Чим більшим ми вкладаємо валюти X тим більшим стає його відношення із Y, тим самим чим більше існує валюти X тим менш вона ціна відносно Y за мінімальну одиницю, тим самим AMM біржі балансують ціни.

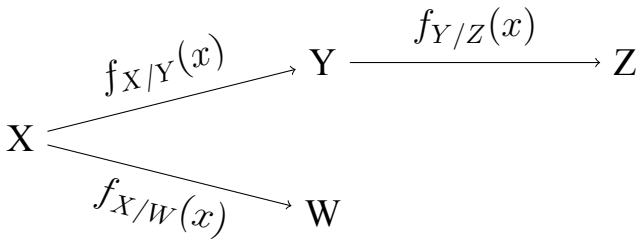

Рис. 3: *Графічне представлення графу біржи*

Більшість сучасних алгоритмів знаходження оптимального шляху між всіма парами (all-to-all по класифікації [5]) потребують аби вага ребра була дійсною функцією від двох вершин $f : (V, V) \to \mathbb{R}$, проте у нашому випадку розглядаєма вага ребра залежить від теперішніх вкладів у вершини та об'єму обміну. Спробуємо розглянути можливі випадки переведення нашої задачі до стандартної із можливістю "сумувати" ваги, та порівнювати їх між собою.

3.1. Обчислення ваги при фіксованому об'ємі

Зафіксуємо стан графу при деякому t, таким чином розміри вкладів у функції обміну стають константами, проте об'єми обміну все ще є невідомим значенням. Тому нехай для пари X/Y зафіксуємо об'єм обміну як відоме x.

Таким чином всі функції обміну на будь-якому з можливих шляхів між X та Y обчислюються у дійсні значення з $\mathbb{R}^+$ за допомогою, наприклад, (7). Проте при обході графа утворенні значення неможна порівнювати допоки шлях з X в Y не дійде докінця.

Нехай стан біржи на момент t має вигляд як на 4. Для знаходження "найкоротшого" шляху (у нашому випадку шляху при котрому ми за X отримаємо якомога більше Y) у подібному графі здавалося б можна використати, наприклад, алгоритм Дейкстри [6] так як на цей раз всі значення ваг є просто числами. Проте, якщо розгля-

нути $S_{X/W}$ та $S_{X/Z}$, то достатньо легко зрозуміти, що значення об'єму цих обмінів не можна порівнювати так як вони є зовсім різними валютами.

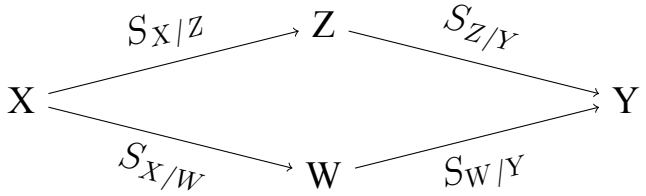

Рис. 4: *Графічне представлення двох можливих шляхів обміну між X та Y при фіксованому x. $S_{X/Y}$ — це значення отримуємої валюти Y за фіксонану кількість x валюти X*

Не рідко на біржах існує валюта до котрої можна перевести будь яку іншу на платформі. Наприклад на фіатних біржах чи біржах ціних паперів — це може бути долар, а на криптобіржах нативна валюта мережі (ethereum, bitcoin).

Лема 3. *Тому пропонується розглянути два припущення:*

- *Якщо в такій ситуації існує пара обміну між W та Z, то через неї ми можемо привести номінал одного значення в інший аби порівняти значення.*
- *Якщо на біржі присутня "базова" валюта, до котрої можна перевести будь-яку іншу, то можна перевести обидві валюти в одну "базову" для порівняння.*

Таким чином послідовність дій для порівняння ребер можна описати алгоритмом 1. Приклад графу після алгортму є рисунок 5.

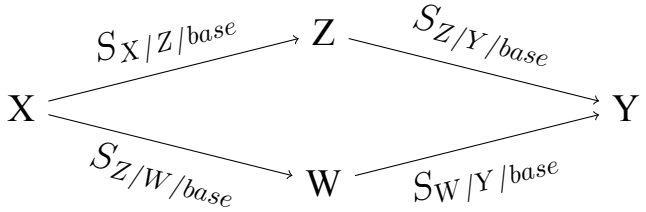

Рис. 5: *Графічне представлення двох можливих шляхів обміну між X та Y при фіксованому x. S_{base} — значення отримуєме при обміні що переведенне до базових валюти*

Для переведення задачі до стандартної для використання алгоритмів обходу графа достатньо за вагу обчислити різницю минулого отриманого значення і теперішнього (рис. 6). Тобто, якщо S_i — об'єми призведені до базових на шляху між валютами, то ваги будуть:

$$W_i = S_i - S_{i-1}, S_0 = 0$$

Algorithm 1 Алгоритм знаходження ваги ребра

function swap(Δx, X, Y) ▷ Об'єм обміну між вершинами X та Y
end function
function choose(X) ▷ Вибрати як наступний шлях обміну X
end function
Ensure: V_t ▷ множина вершин графа
Ensure: E_t ▷ множина ребер
Ensure: $v_{i+1} \in V_t$ ▷ перша вершина через котру проходить шлях
Ensure: $v_{j+1} \in V_t$ ▷ друга вершина з котрою порівнюється шлях
Ensure: $v_{base} \in V_t$ ▷ вершина базової валюти біржі
Require: $\Delta x > 0$
$volume_{left} \leftarrow 0$
$volume_{right} \leftarrow 0$
if $(v_{i+1}, v_{j+1}) \in E_t$ **then**
$volume_{tmp} \leftarrow$ swap$(\Delta x, v_i, v_{i+1})$
$volume_{left} \leftarrow$ swap$(volume_{tmp}, v_{i+1}, v_{j+1})$
$volume_{right} \leftarrow$ swap$(\Delta x, v_j, v_{j+1})$
else if $(v_{base}, v_{i+1}) \in E_t \wedge (v_{base}, v_{j+1}) \in E_t$ **then**
$volume_{left} \leftarrow$ swap$(\Delta x, v_{base}, v_{i+1})$
$volume_{right} \leftarrow$ swap$(\Delta x, v_{base}, v_{j+1})$
end if
if $volume_{left} < volume_{right}$ **then** choose(v_{j+1})
else choose(v_{i+1})
end if

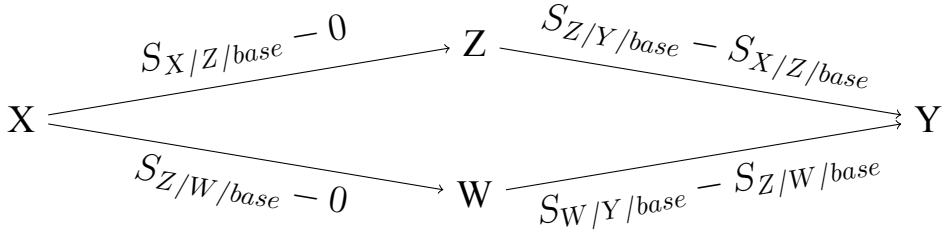

Рис. 6: *Графічне представлення двох можливих шляхів обміну між X та Y при фіксованому x після переведення задачі до стандартної*

3.2. Преоптимізації

4. Імплементація програми

У цьому розділі ми розглянемо структуру програми що імлементує алгоритм пошуку оптимального шляху між вершинами біржевого графа на прикладі UniswapV2 в мережі Ethereum [10].

4.1. Об'єкти та сутності програми

У цьому розділі показана спрощена діаграма взаємозв'язків даних та сутностей, що використовує програма.

4.1.1. Блоки

Події в криптовалютній мережі Ethereum є послідовними, і збираються учасниками в блоки що йдуть один за одним. Кожен блок у мережі починаючи з першого має свій порядковий номер, що також називається "висотою" блока.

Так як зміни стану графа залежать від подій у блоках, замість непервного часу t імплементація використовує дискретне значення висоти блоку $t \in \mathbb{N}_0$ у котрому сталося оновлення даних, що в деяких випадках значно спрощує імлементацію і дозволяє прив'язати дані до конкретної висоти.

4.1.2. Токени

Токен (з англ. *token*) — є смартконтрактом стандарту ERC20 [9] сукупністю інформації про валюту в криптобіржі, починаючи з її унікальної адреси в мережі, закінчуючи її назвою та коротким символьним кодом. Фактично сукупність токенів в программі це множина вершин біржевого графа E_t, де унікальність валюти забезпечується унікальністю адреси в мережі.

4.1.3. Пари

Пара — це сматрконтракт, що створюється біржою на конкретному блоці як зв'язок двух токенів, і що тримає в собі ліквідність вкладників (резерви) для створення обмінів. Так як це зв'язок двух токенів, сукупність всіх пар є множиною ребер V_t з вершин E_t.

Аналогічно резерви є вагою ребер, ліквідністю у токенах з обох сторін пари, що зберігається як пара чисел.

4.1.4. Діаграма зв'язків

Приклад створення бази даних за допомогою SQL можна побачити у ДОДАТКУ А 5.

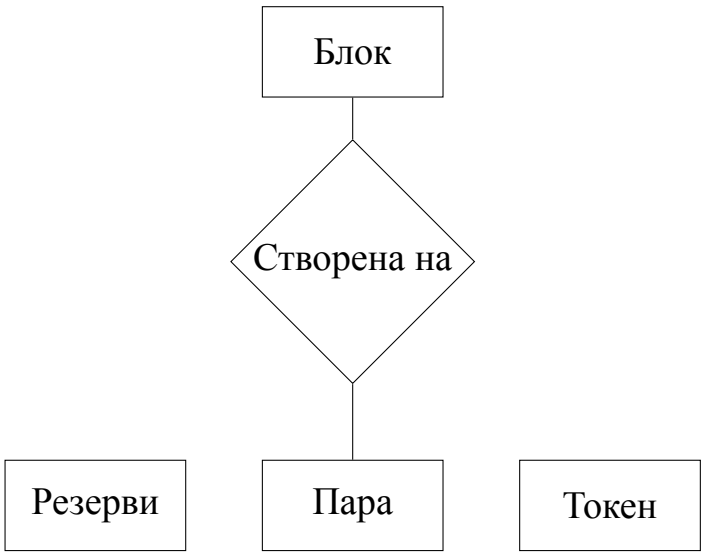

Рис. 7: *Спрощена ER діаграма зв'язків у БД*

4.2. Збір данних

Перед початком роботи програмі необхідно зібрати всі необхідні дані

5. ВИСНОВОКИ

Література

[1] Hayden Adams та ін. “Uniswap v3 core”. В: *Tech. rep., Uniswap, Tech. Rep.* (2021).

[2] Guillermo Angeris та Tarun Chitra. “Improved Price Oracles: Constant Function Market Makers”. В: *Proceedings of the 2nd ACM Conference on Advances in Financial Technologies*. AFT ’20. ACM, жовт. 2020. doi: 10.1145/3419614.3423251. url: http://dx.doi.org/10.1145/3419614.3423251.

[3] Guillermo Angeris та ін. “The Geometry of Constant Function Market Makers”. В: (серп. 2023).

[4] Vincent Danos, Hamza El Khalloufi та Julien Prat. “Global Order Routing on Exchange Networks”. В: *FC 2021: Financial Cryptography and Data Security. FC 2021 International Workshops*. За ред. Matthew Bernhard та ін. Т. 12676. Lecture Notes in Computer Science. Virtual Event, France: Springer, бер. 2021, с. 207—226. doi: 10.1007/978-3-662-63958-0_19. url: https://hal.science/hal-03455981.

[5] N. Deo, C. Pang та United States. Department of Transportation. *Shortest Path Algorithms: Taxonomy and Annotation*. Washington State University, Computer Science Department, 1980. url: https://books.google.com.ua/books?id=S91_GwAACAAJ.

[6] E. W. Dijkstra. “A note on two problems in connexion with graphs”. В: *Numer. Math.* 1.1 (груд. 1959), с. 269—271. issn: 0029-599X. doi: 10.1007/BF01386390. url: https://doi.org/10.1007/BF01386390.

[7] John McCarthy. “MEASURES OF THE VALUE OF INFORMATION”. В: *Proceedings of the National Academy of Sciences* 42.9 (1956), с. 654—655. doi: 10.1073/pnas.42.9.654. eprint: https://www.pnas.org/doi/pdf/10.1073/pnas.42.9.654. url: https://www.pnas.org/doi/abs/10.1073/pnas.42.9.654.

[8] D.D. Šiljak. “Dynamic graphs”. В: *Nonlinear Analysis: Hybrid Systems* 2.2 (2008). Proceedings of the International Conference on Hybrid Systems and Applications, Lafayette, LA, USA, May 2006: Part II, с. 544—567. issn: 1751-570X. doi: https://doi.org/10.1016/j.nahs.2006.08.004. url: https://www.sciencedirect.com/science/article/pii/S1751570X07000738.

[9] Fabian Vogelsteller та Vitalik Buterin. “ERC-20: Token Standard”. В: *Ethereum Improvement Proposals* 20 (листоп. 2015). [Online serial]. url: https://eips.ethereum.org/EIPS/eip-20.

[10] Gavin Wood. “Ethereum: A secure decentralised generalised transaction ledger”. В: ().

[11] Yi Zhang, Xiaohong Chen та Daejun Park. “Formal specification of constant product (xy= k) market maker model and implementation”. B: *White paper* (2018).

ДОДАТОК А

У цьому додатку наведений приклад SQL коду для оформлення бази даних у імплементації даного алгоритму.

Таблиця блоків `blocks`:

```
CREATE TABLE blocks (
    height BIGINT   PRIMARY KEY,
    hash   CHAR(66) NOT NULL UNIQUE
);
```

Таблиця токенів `tokens`:

```
CREATE TABLE tokens (
    id SERIAL PRIMARY KEY,

    address CHAR(42) NOT NULL UNIQUE,

    name   TEXT NOT NULL,
    symbol TEXT NOT NULL,
);
```

Таблиця пар `pairs`:

```
CREATE TABLE IF NOT EXISTS pairs (
    id      SERIAL   PRIMARY KEY,
    address CHAR(42) NOT NULL UNIQUE,
    token0  INTEGER  NOT NULL,
    token1  INTEGER  NOT NULL,
    block  INTERGER NOT NULL,

    FOREIGN KEY (token0) REFERENCES tokens(id),
    FOREIGN KEY (token1) REFERENCES tokens(id),
    FOREIGN KEY (block)  REFERENCES blocks(height),
);
```

Таблиця резервів `reserves`:

```
CREATE TABLE IF NOT EXISTS reserves (
    id    SERIAL  PRIMARY KEY,
    pair  INTEGER NOT NULL,
```

```
    block INTEGER NOT NULL,

    reserve0 NUMERIC NOT NULL,
    reserve1 NUMERIC NOT NULL,

    FOREIGN KEY (pair)  REFERENCES pairs(id),
    FOREIGN KEY (block) REFERENCES blocks(height)
);
```