ВСТУП ДО АЛГОРИТМІВ

СУЛЕМА Ольга Костянтинівна, PhD
email: sulema.olga@lll.kpi.ua
Telegram: @olga_sulema

Лекція №1 від 4 вересня 2024 р.

Як приготувати чашку чаю?

Джерело зображення: https://fabriory.com.ua/sites/default/files/styles/large/public/13-152_1.jpg?itok=BpbJCYiS

Як приготувати чашку чаю?

Для того, щоб приготувати чашку чаю, потрібно:

- Покласти в чашку заварку або пакетик чаю
- Залити воду у чайник
- Закип’ятити воду
- Залити окріп у чашку
- Додати додаткові інгредієнти (цукор, молоко, спеції тощо)
- Розмішати чай

**Це рецепт, тобто послідовність інструкцій,
що, фактично, є алгоритмом!**

Що таке алгоритм?

Таким чином…

Алгоритм – це скінченна послідовність однозначних дій, яку необхідно виконати для розв’язання поставленої задачі та досягнення очікуваного результату.

Джерело зображення: https://www.alphabetsecretarial.co.uk/wp-content/uploads/transcription-steps.jpg

Заглибимось у історію…

- У IX сторіччі перський математик **Аль-Хорезмі** написав трактат “*Коротка книга доповнення і протиставлення*”.
  - Він описав, серед іншого, правила додавання, множення та ділення чисел, заклавши основи класичної алгебри.
  - Це були ***алгоритми***.
- Саме слово **алгоритм** походить від латинізованого імені **Аль-Хорезмі**.

Мухаммад ібн Муса аль-Хорезмі

Джерело зображення: https://steemitimages.com/p/23KQwnti57stwuABfWdbtmEH3RUDMj8vPJ91EKtMnJCrtAHhjTrBh7KMuCsCznbNXjykeTiPurMXV5Kt7pi6Ya92B8iEdZ5?format=match&mode=fit&width=640

Ще глибше у тисячоліття…

Але все почалось набагато раніше…

- Найдавніший відомий алгоритм – це **алгоритм ділення**, знайдений на шумерський глиняній табличці (близько 2500 року до н.е.)
- Одним з найдавніших алгоритмів, який використовується до сьогодні, вважається **алгоритм Евкліда** (близько 300 року до н.е.)
- Інший стародавній алгоритм – так зване **решето Ератосфена** (близько 100 року до н.е.)

Властивості алгоритму

Алгоритм повинен відповідати наступним властивостям:

- **Точність**
  - Алгоритм має містити однозначно визначену послідовність однозначно сформульованих дій (кроків).
- **Скінченність**
  - Алгоритм має завершитися після виконання визначеної кількості кроків за скінченний проміжок часу.
- **Здійсненність**
  - Алгоритм має бути здійсненим в умовах реального світу. Він не може бути абстрактним або уявним.

Властивості алгоритму

Алгоритм повинен відповідати наступним властивостям:

- **Результативність**
  - Алгоритм завжди повинен завершуватись отриманням кінцевого результату, який має відповідати поставленій задачі.
- **Масовість**
  - Алгоритм має бути застосованим до всього класу задач такого типу, які відрізняються лише початковими даними.
- **Ефективність**
  - Алгоритм повинен забезпечувати розв’язання задачі за мінімальний час із мінімальними ресурсами.

Джерело зображення: http://www.manassaslawyers.com/wp-content/uploads/2013/08/question-11.jpg

Навіщо потрібно вивчати алгоритми?

Розглянемо наступні задачі…

- Компанія, яка надає послуги з трансферу до аеропорту, повинна скласти розклад, щоб доправити всіх клієнтів вчасно. *Який шатл кого забере та в якій послідовності?*
- Кур'єрська служба має багато замовників та декілька вантажівок, які перевозять товари. *Як потрібно спланувати доставку до замовників, щоб мінімізувати витрати служби?*

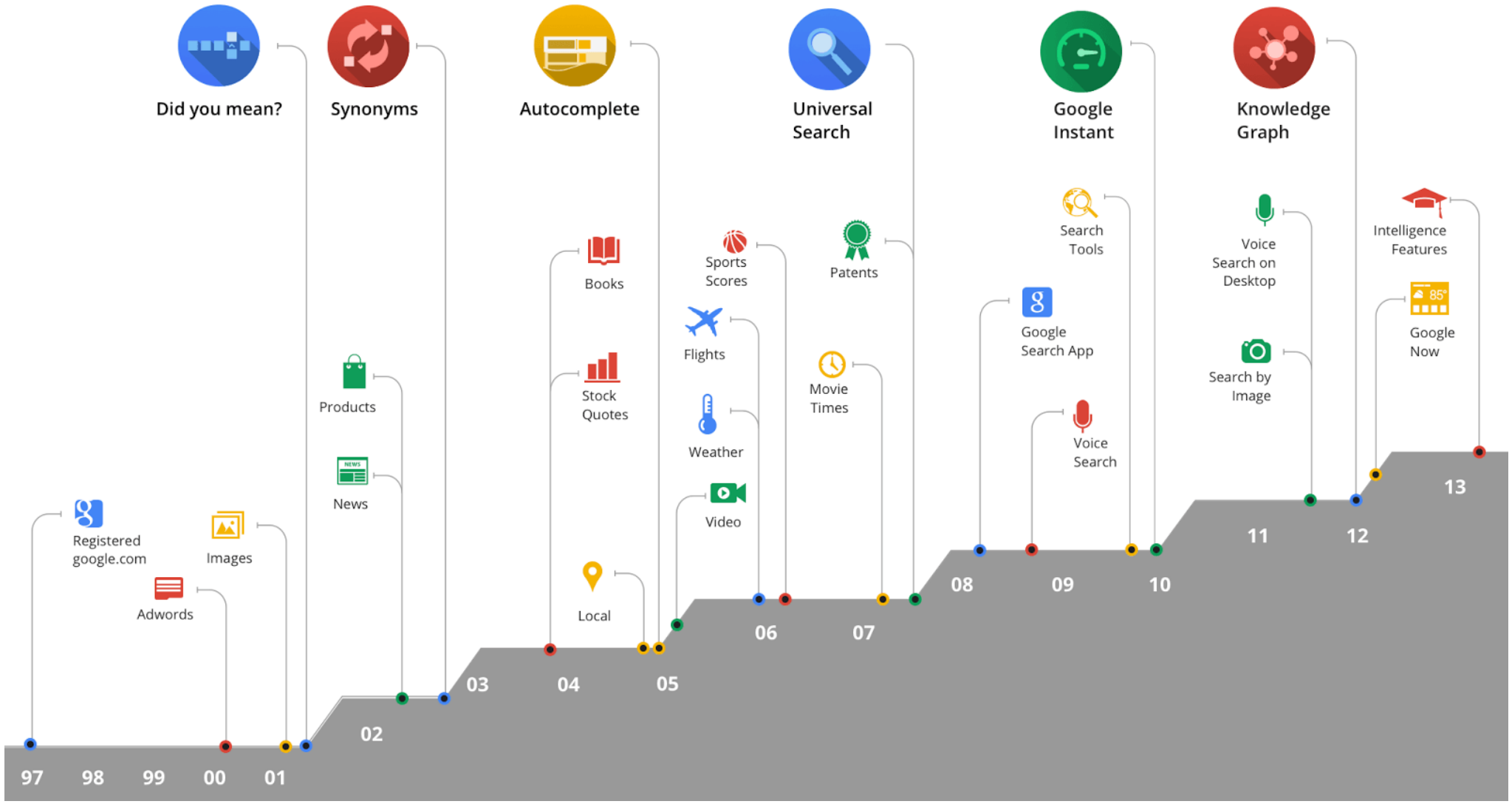
Існує багато можливих варіантів!

- Деякі з них є дешевими, інші можуть бути надто дорогими або просто небажаними.
- **Алгоритми допоможуть нам визначити найкращий варіант!**

Деякі важливі застосування алгоритмів

ПРИКЛАДИ

Google Search Timeline

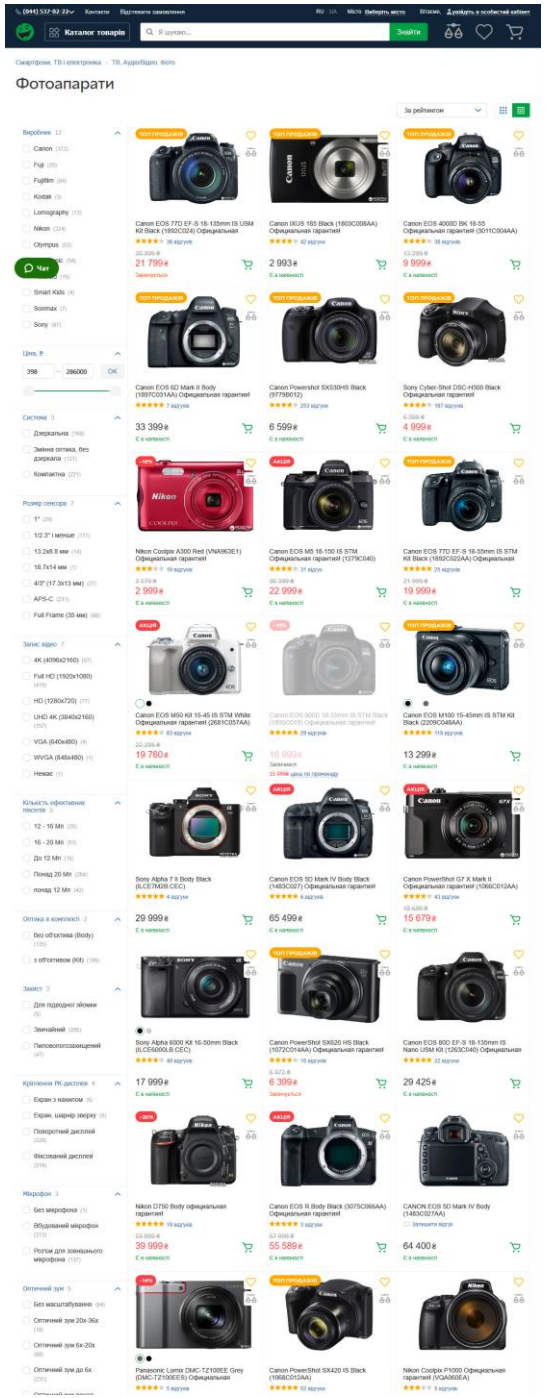


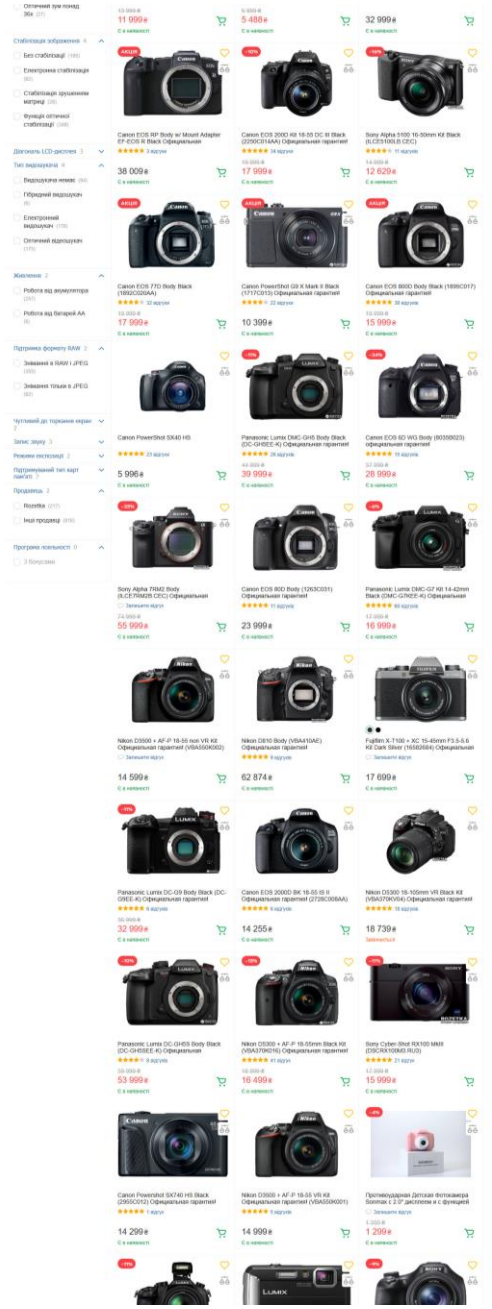
Джерело зображення: https://www.woveon.com/wp-content/uploads/2018/06/how-google-ranks-your-business-works.png

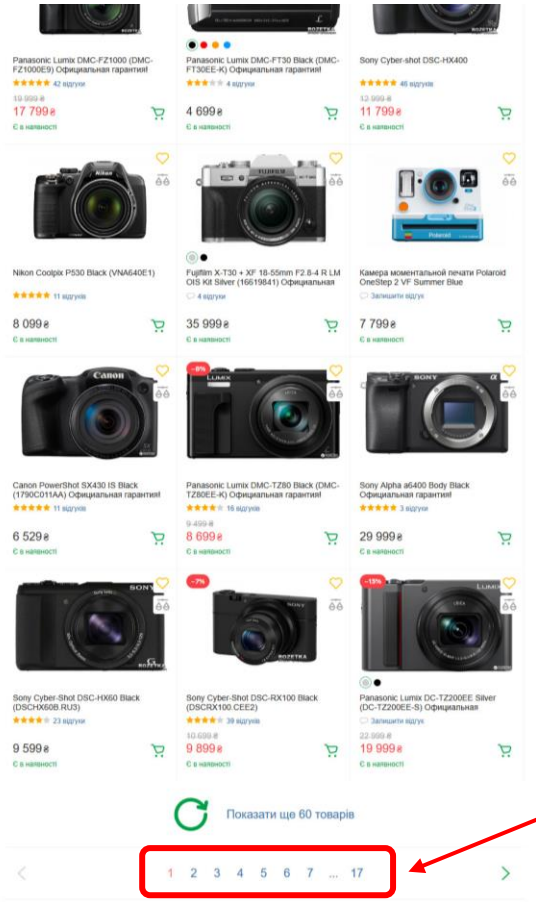
Деякі важливі застосування алгоритмів

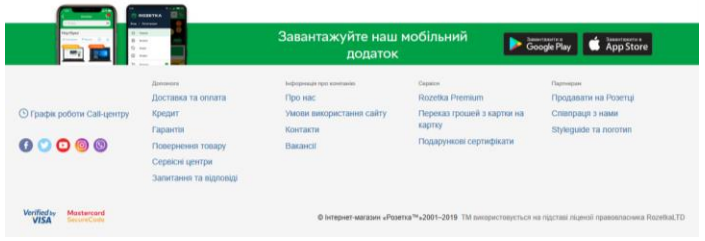
1. Пошук (Searching)

- Пошук інформації та підтвердження того, що ця інформація є саме тою, що необхідна – вкрай важлива задача сьогодення. Без можливості пошуку найбанальніші дії, які ми виконуємо онлайн, були б просто нездійсненими. Наприклад, не існував би Google.







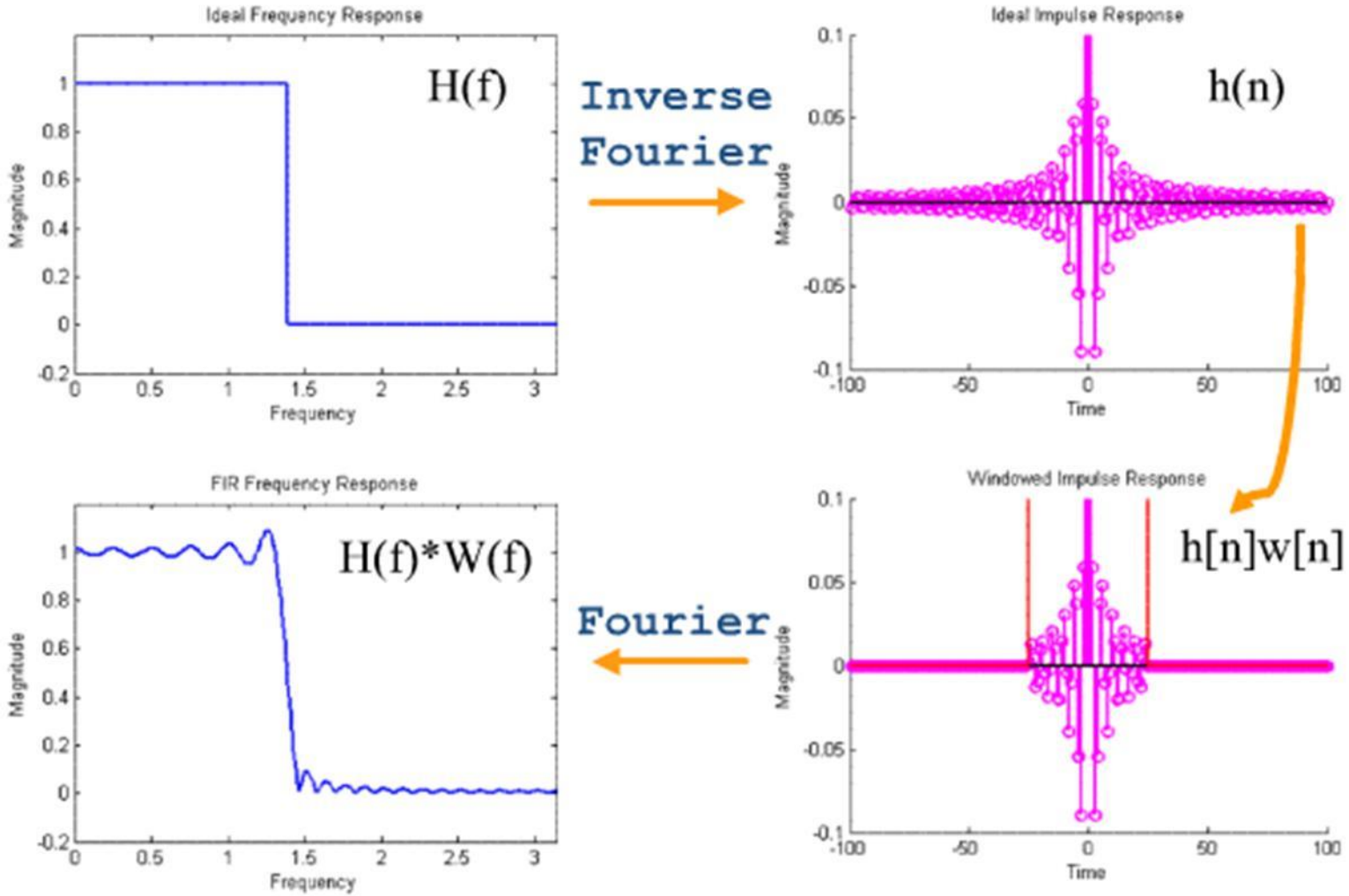


Джерело зображення:https://icon-library.net/images/stress-icon/stress-icon-4.jpg

Деякі важливі застосування алгоритмів

2. Сортування (Sorting)

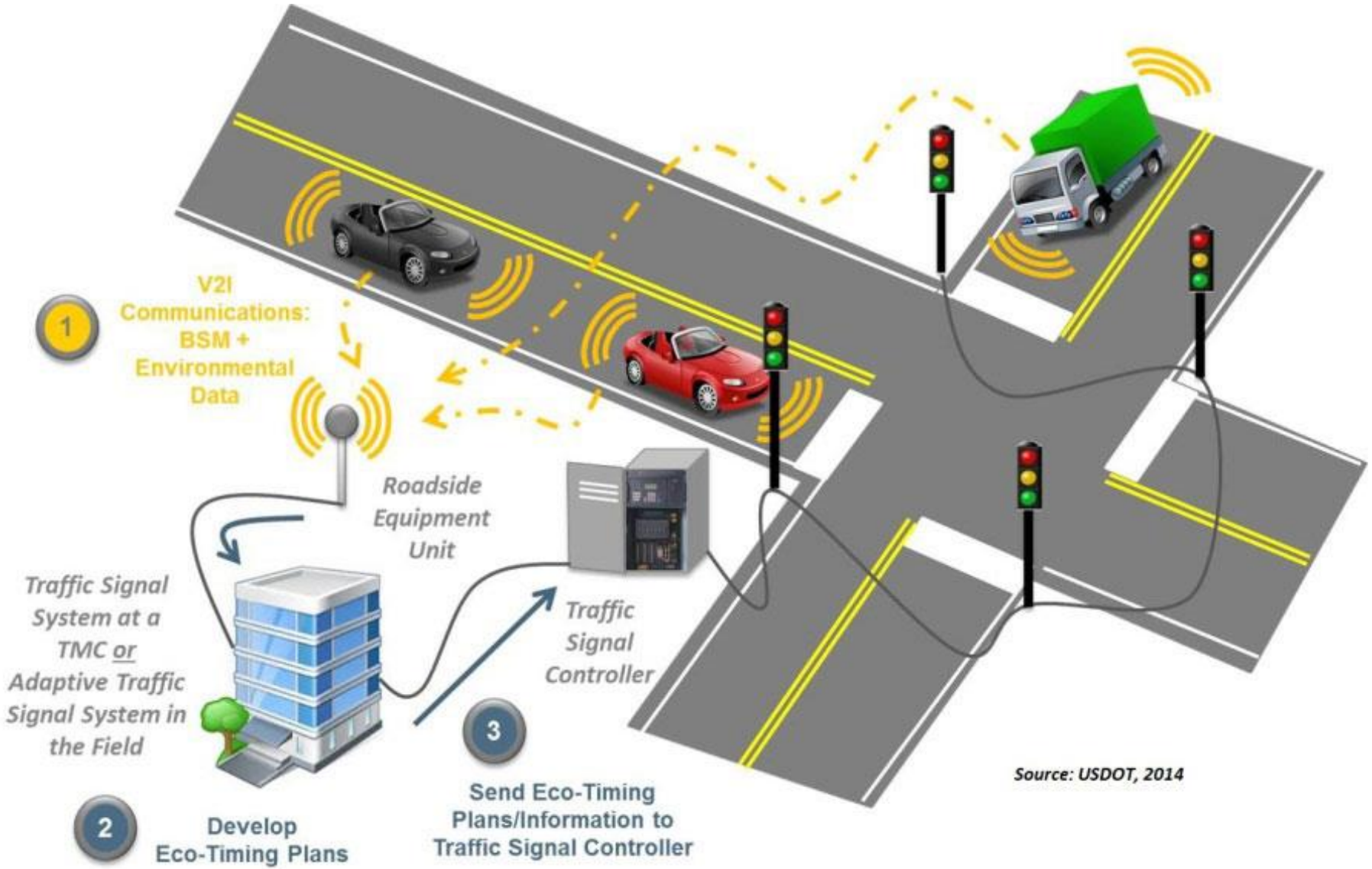
- В сучасному світі надмірної інформації впорядкування отриманих даних відповідно до потреб конкретної людини рятує від інформаційного перевантаження.
- Крім того, досить багато **складних алгоритмів** потребують, щоб дані були спочатку відсортовані певним чином.


Ideal Frequency Response
H(f)
Magnitude
Frequency
Inverse
Fourier
Ideal Impulse Response
h(n)
Magnitude
Time
Windowed Impulse Response
h[n]w[n]
Magnitude
Time
Fourier
FIR Frequency Response
H(f)*W(f)
Magnitude
Frequency

Деякі важливі застосування алгоритмів

3. Перетворення даних (Transforming)

- Перетворення даних одного типу у дані іншого типу є важливою задачею для їх ефективного використання.
- Наприклад, швидке перетворення Фур’є (FFT) дозволяє перетворювати сигнали з часової області у частотну, завдяки чому спрощується цифрова фільтрація голосового сигналу у телекомунікаційних системах.

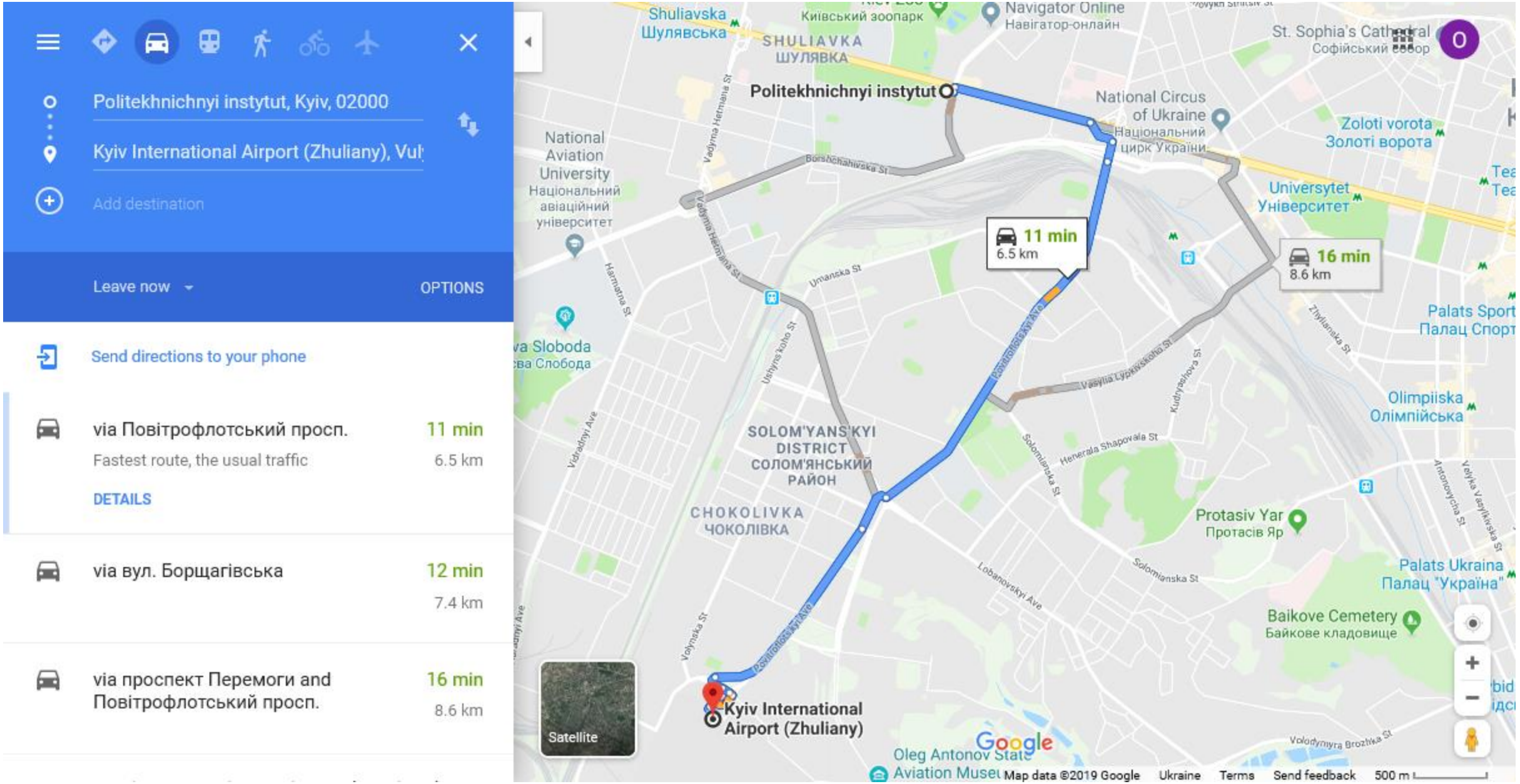


Джерело зображення: https://www.itsknowledgeresources.its.dot.gov/its/bcllupdate/ArterialTrafficControl/image2.jpg

Деякі важливі застосування алгоритмів

4. Планування (Scheduling)

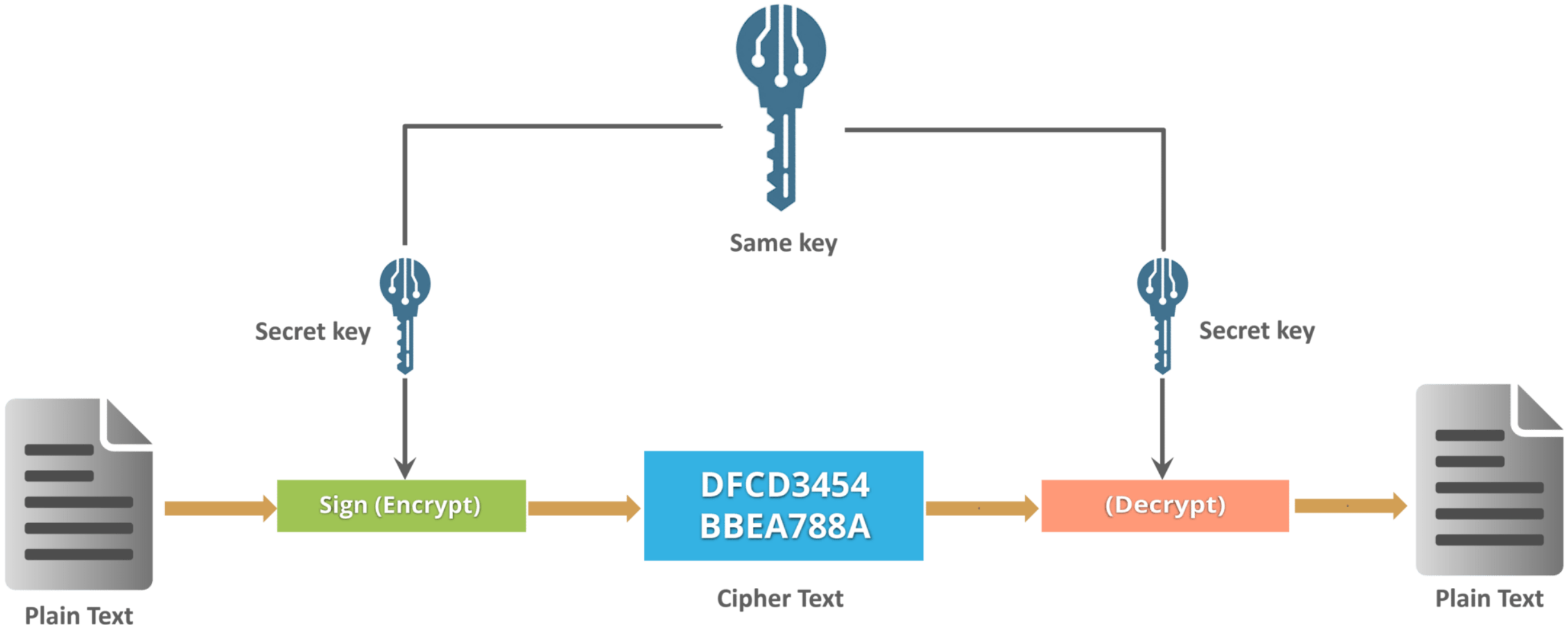
- Задачу планування неможливо було б розв’язати без використання алгоритмів.
- Зокрема, однією з ключових концепцій багатозадачності у багатопроцесорних системах є планування виконання завдань, яке полягає в призначенні пріоритетів процесам у черзі з пріоритетами.


Politekhnichnyi instytut, Kyiv, 02000
Kyiv International Airport (Zhuliany), Vul
Add destination
Leave now
OPTIONS
Send directions to your phone
via Повітрофлотський просп.
11 min
Fastest route, the usual traffic
6.5 km
DETAILS
via вул. Борщагівська
12 min
7.4 km
via проспект Перемоги and Повітрофлотський просп.
16 min
8.6 km
Politekhnichnyi instytut
Kyiv International Airport (Zhuliany)
11 min
6.5 km
16 min
8.6 km
Satellite
SHULIAVKA
ШУЛЯВКА
SOLOM'YANS'KYI DISTRICT
СОЛОМ'ЯНСЬКИЙ РАЙОН
CHOKOLIVKA
ЧОКОЛІВКА
National Aviation University
Національний авіаційний університет
National Circus of Ukraine
Національний цирк України
Universytet
Університет
Protasiv Yar
Протасів Яр
Baikove Cemetery
Байкове кладовище
Palats Ukraina
Палац "Україна"
Olimpiiska
Олімпійська
Palats Sport
Палац Спорт
Zoloti vorota
Золоті ворота
Map data ©2019 Google
Ukraine
Terms
Send feedback
500 m

Деякі важливі застосування алгоритмів

5. Аналіз графів (Graph analysis)

- Алгоритми на графах широко використовується в найрізноманітніших галузях людської діяльності. Наприклад, принцип роботи GPS базується саме на пошуку найкоротших шляхів між двома визначеними точками.

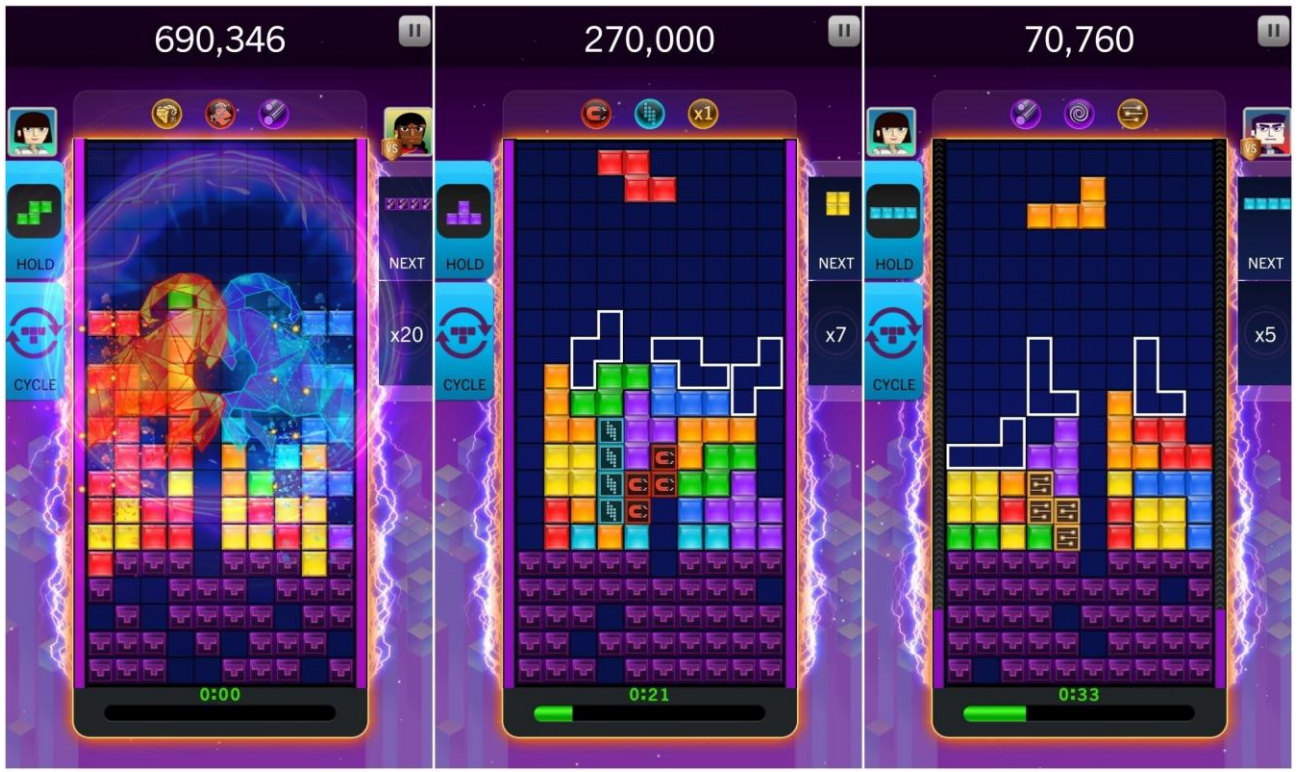


Джерело зображення: https://d1jnx9ba8s6j9r.cloudfront.net/blog/wp-content/uploads/2018/07/symmetric-key-what-is-cryptography-edureka.png

Деякі важливі застосування алгоритмів

6. Криптографія (Cryptography)

- Захист даних – одна з найактуальніших задач сьогодення. Алгоритми шифрування даних запобігають несанкціонованому доступу до них із можливістю відновлення закодованої інформації у первісному вигляді.



Джерела зображень: http://www.expertchessstrategies.com/images/ziccidus chess.jpg
https://s3.amazonaws.com/tetris-www/assets/article/2018/09/10/recover-misplaced-tetrimino_feature.jpg

Деякі важливі застосування алгоритмів

7. Псевдовипадкова генерація чисел

- Уявіть гру, яка завжди працює однаково. Ви завжди починаєте з того ж самого місця, виконуєте ті ж самі кроки таким самим шляхом.
- Без можливості генерувати випадкові на перший погляд числа не тільки ігри, але й деякі інженерно-технічні задачі були б нездійснені.

Отже, навіщо ми вивчаємо алгоритми?

Джерела зображень: https://www.digitaltechnologieshub.edu.au/images/default-source/topics/tile-images/algorithm.jpg

Навіщо вивчати алгоритми?

- **Алгоритми використовуються в усіх галузях комп’ютерних наук**
  - Протоколи маршрутизації в **мережах зв’язку** створюються за допомогою класичних алгоритмів найкоротшого шляху
  - **Криптографія** з відкритим ключем спирається на ефективні алгоритми теорії чисел
  - **Комп'ютерна графіка** вимагає геометричних примітивів, які генеруються алгоритмічно
  - **Бази даних** використовують алгоритми пошуку на деревах
  - В **обчислювальній біології** використовуються алгоритми динамічного програмування для вимірювання подібності геномів
  - … *Цей список можна продовжувати …*

Навіщо вивчати алгоритми?

- Алгоритми використовуються в усіх галузях комп’ютерних наук
- **Алгоритми є рушіями технологічних інновацій**

Джерело зображення: https://cdn.vox-cdn.com/thumbor/Pkmq1nm3skO0-j693JTMd7RL0Zk=/0x0:2012x1341/1200x800/filters:focal(0x0:2012x1341)/cdn.vox-cdn.com/uploads/chorus_image/image/47070706/google2.0.0.jpg

Навіщо вивчати алгоритми?

- Алгоритми використовуються в усіх галузях комп’ютерних наук
- Алгоритми є рушіями технологічних інновацій
- **Алгоритми дають новий погляд на інші науки**
  - Дослідження **квантових алгоритмів** дозволило переосмислити погляди на квантову механіку
  - Коливання цін на **економічних ринках** можна розглядати як алгоритмічний процес
  - **Еволюція** може розглядатись як вражаюче ефективний пошуковий алгоритм

Навіщо вивчати алгоритми?

- Алгоритми використовуються в усіх галузях комп’ютерних наук
- Алгоритми є рушіями технологічних інновацій
- Алгоритми дають новий погляд на інші науки
- **Алгоритми розвивають мислення**

Джерело зображення: https://researchdigest.files.wordpress.com/2016/11/thinkstockphotos-597273530.jpg

Навіщо вивчати алгоритми?

- Алгоритми використовуються в усіх галузях комп’ютерних наук
- Алгоритми є рушіями технологічних інновацій
- Алгоритми дають новий погляд на інші науки
- Алгоритми розвивають мислення
- **Алгоритми – це цікаво!**

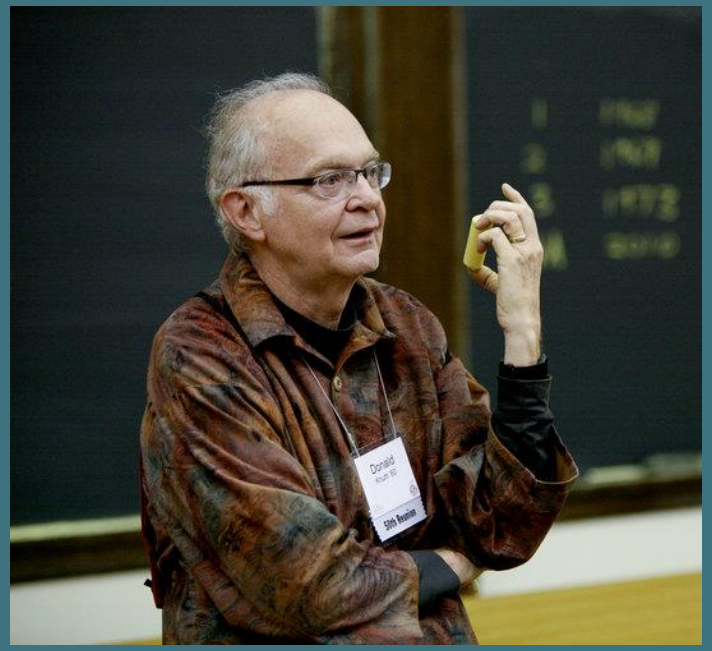
Джерело зображення: https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcQpd_QfIPZLh47QrQoh7Bk9yMLrhi43YngJbrvLkxwAJxD2zJKe4g

Способи представлення алгоритмів

ЯКИМ ЧИНОМ МОЖЕ БУТИ ПРЕДСТАВЛЕНИЙ АЛГОРИТМ

Джерело зображення: https://amp.businessinsider.com/images/55a003fb6bb3f7996df7425c-750-500.jpg

ДОНАЛЬД КНУТ



An algorithm must be seen to be believed.

Алгоритм необхідно побачити, щоб у нього повірити.

Як можна описати алгоритм?

- Словесно
- Аналітично
- Графічно
- Програмно

Як можна описати алгоритм?

Словесний опис

- ✓ Може бути легко написаний будь-ким
- ✓ Є непоганим варіантом опису простих інструкцій, які потрібно виконувати у лінійному порядку
- ✗ Природна мова може бути неоднозначною
- ✗ Важко формалізувати параметри та характеристики
- ✗ Опис може бути достатньо довгим
- ✗ Описати словами складну логіку може бути дуже складно
- ✗ Спосіб запису є невізуальним

Приклад

Алгоритм: пошук найбільшого спільного дільника (НСД) двох чисел

1. Якщо перше число дорівнює нулю, друге є НСД, інакше перейти до кроку 2.
2. Якщо друге число дорівнює нулю, перше є НСД, інакше перейти до кроку 3.
3. Якщо перше число більше за друге, поділити його на друге число та перейти до кроку 6, інакше перейти до кроку 4.
4. Якщо друге число більше за перше, поділити його на перше число та перейти до кроку 6.
5. Якщо перше число дорівнює другому, будь-яке з них є НСД, інакше перейти до кроку 6.
6. Якщо остача від ділення дорівнює нулю, дільник з попереднього кроку є НСД, інакше перейти до кроку 7.
7. Якщо остача від ділення не дорівнює нулю, ділене з попереднього кроку замінити на остачу від ділення та перейти до кроку 3.

Як можна описати алгоритм?

Аналітичний (словесно-формульний) опис

- ✓ Можна формально визначити параметри та характеристики, використовуючи формули та позначення
- ✓ Є непоганим способом при алгоритмізації фізико-математичних задач
- ✓ Легко зрозумілий для експертів з галузі знань, що алгоритмізується
- × Може вимагати знань зі специфічних галузей
- × Використовується природна мова, якій притаманна неоднозначність
- × Описати складну логіку може бути доволі складно
- × Спосіб запису є невізуальним

Приклад

Алгоритм: пошук найбільшого спільного дільника (НСД) двох чисел

Нехай a та b – цілі числа, не рівні одночасно нулю, і послідовність чисел $a > b > r_1 > r_2 > \cdots > r_n$ визначена тим, що кожне r_i – це остача від ділення передпопереднього числа на попереднє, а передостаннє ділиться на останнє націло:

$$a = bq_0 + r_1$$
$$b = r_1q_1 + r_2$$
$$r_1 = r_2q_2 + r_3$$
...
$$r_{n-1} = r_nq_n$$

Тоді НСД(a, b) дорівнює r_n, останньому ненульовому члену цієї послідовності.

Література

- *Algorithms. Design and Analysis*, Harsh Bhasin, Oxford University Press
- *Algorithms For Dummies*, John Paul Mueller and Luca Massaron
- *Algorithms Illuminated. Part 1: The Basics*, Tim Roughgarden, Stanford University
- *Whom to marry, how to cook and where to buy gas: solving dilemmas of daily life, one algorithm at a time*, Samir Khuller, University of Maryland