

## 1) Збір даних

Для проведення дослідження необхідно накопичити дані про виконавців, їхні альбоми, а згодом пісні з цих альбомів.

Під час збору інформації її треба буде зберігати. Дані(пісні, альбоми, виконавці) будуть представлені як об'єкти, які мають певні властивості, зокрема унікальний id.

Перелік операцій, які повинен підтримувати ADT:

- 1) Кожний унікальний елемент в контейнері не повинен повторюватись двічі
- 2) Додати елемент до контейнеру
- 3) Перевірити елемент на входження в контейнер
- 4) Дізнатися кількість елементів в контейнері
- 5) Порядок елементів не важливий

Set adt чудово підходить під усі задачі, адже кожен елемент в set – унікальний.

Реалізуючи структуру даних множина, візьму за основу тип даних dict в python.

По-перше, пара ключ – значення дасть змогу перевіряти елемент на входження, знаючи лише його id. По-друге, dict в python реалізовано на основі хеш-таблиці, що дає хорошу оцінку на вставку елемента та перевірки його на входження (обидві  $O(1)$ ). По-третє, ми можемо однозначно задати хеш кожного об'єкту, який будемо зберігати, оскільки кожен елемент має свій унікальний id.

## 2) Аналіз даних

Під час аналізу даних вже не можна нехтувати порядком, оскільки нам важлива популярність кожної пісні.

Перелік операцій, який повинен підтримувати adt

- 1) Додати елемент до контейнеру
- 2) Дізнатися кількість елементів в контейнері
- 3) Проітерувати від найбільшого елемента до найменшого

Зазначу, що нам не важлива унікальність елементів, оскільки ми вважаємо, що взяли їх з set, де елемент не може повторюватись двічі.

Вибір стоїть між array adt та list adt. Вважаю, що array adt краще підійде під ці вимоги, оскільки можна приблизно оцінити кількість пісень кожного року і відразу задати фіксований розмір масиву (даних не так багато, щоб переживати за оперативну пам'ять). Тоді ми зможемо одразу доступатися до елемента за індексом

Також список важко важче сортувати та ми все одно будемо зберігати кожен наступний елемент, що зробить витрати пам'яті приблизно рівними, порівнюючи з масивом (враховуючи невелику кількість елементів).

Я використаю реалізацію масиву на основі виділення пам'яті, використовуючи ctypes, що дозволить зберігати в масиві екземпляри будь-якого класу.