

HW #02: Inverted Index CLI

1. Описание задания	2
1.1. Инвертированный индекс (Inverted Index)	2
1.2. Входные данные	2
1.3. Требования к реализации	2
2. Рекомендации	5
3. Критерии оценивания	6
4. Инструкция по отправке задания	6
5. FAQ (часто задаваемые вопросы)	9
6. Дополнительные задания (не на оценку)	10



1. Описание задания

В этом задании вам нужно написать консольный интерфейс к библиотеке по работе с инвертированным индексом. Цель задания - завести привычки:

1. Аннотировать код, писать документацию и выбирать лаконичные имена методов и функций (naming);
2. Читать официальную документацию с целью поиска релевантной функциональности.

1.1. Инвертированный индекс (Inverted Index)

Вводные про инвертированный индекс представлены в описании задания "Inverted Index Lib" (учебный модуль про pytest).

1.2. Входные данные

Дамп Википедии

- Формат: текст
- В каждой строке находятся следующие поля, разделенные знаком табуляции:
 1. INT - id статьи,
 2. STRING - текст статьи,

Пример:

```
12      Anarchism      Anarchism is often defined as a political
philosophy which holds the state to be undesirable, unnecessary, or
harmful.
```

1.3. Требования к реализации

Возьмите за основу решение задания "Inverted Index Lib". На основе решения этого задания библиотека по работе с инвертированным индексом будет предоставлять следующую функциональность:

- `InvertedIndex.query(self, words: List[str]) -> List[int]`
- `InvertedIndex.dump(self, filepath: str) -> None`
- `InvertedIndex.load(cls, filepath: str) -> InvertedIndex`



- `load_documents(filepath: str) -> Dict[int, str]`
индексы документов кастуем к `int`'ам
content не парсим на слова, только строку достаем и отрезаем `"\n"` в конце строк
- `build_inverted_index(load_documents(filepath: str)) -> InvertedIndex`

С учетом примеров и лайфхаков, показанных в рамках этого модуля:

1. Обновите код¹, чтобы он было более лаконичный (см. collections) и не содержал потенциальных ошибок (наследование, значения по умолчанию для сложных объектов и т.п.);
2. Изучите [PEP-257](#) (в отличие от PEP-8 - очень короткий) и дополните вашу библиотеку документацией. Запустите `pylint` **без** указания флагов `"-d invalid-name,missing-docstring"`, и убедитесь, что ошибок и предупреждений нет;
3. Дополните библиотеку консольным интерфейсом, который будет предоставлять возможность:
 - a. Получить подсказку (`help-string`) по его использованию ;
 - b. Построить дамп инвертированного индекса
`load_documents`
→ `build_inverted_index`
→ `InvertedIndex.dump(filepath: str)`
 - c. Использовать дамп инвертированного индекса для обработки поисковых запросов;

Для реализации указанных методов вам будет необходимо обратиться к официальной документации по [argparse](#).

Спецификация на консольный интерфейс:

1. Получение подсказки (`help-string`) должно содержать подстроку `"Inverted Index CLI"`;

```
$ python3 task*_inverted_index_cli.py -h
$ python3 task*_inverted_index_cli.py --help
```

2. Функционал библиотеки и консольного интерфейса должен быть расширяемым, например должна быть предоставлена возможность использовать разные стратегии сохранения инвертированного индекса на жестком диске. Используйте аргумент типа [argparse.add_argument\(choices\)](#), предоставьте возможность выбирать между `json` и `pickle`, значением по умолчанию

¹ В какой шляпе делаем рефакторинг? Красной или зеленой?

используйте "json". Реализовывать pickle необязательно, можно использовать "заглушку" о том, что в текущий момент данная стратегия не реализована.

```
$ python3 task*_inverted_index_cli.py build --strategy json --dataset  
/path/to/dataset --output /path/to/inverted.index  
$ python3 task*_inverted_index_cli.py build --strategy pickle  
--dataset /path/to/dataset --output /path/to/inverted.index  
$ python3 task*_inverted_index_cli.py build --dataset /path/to/dataset  
--output /path/to/inverted.index
```

3. Обработка поисковых запросов. Проверяться будет только json-часть CLI вашей библиотеки. Загрузив инвертированный индекс с жесткого диска вам необходимо предоставить ответы на поисковые запросы. Загружать инвертированный индекс с жесткого диска на один поисковый запрос - дорого, поэтому предоставьте возможность обрабатывать несколько поисковых запросов с помощью указания аргумента `--query` несколько раз. Для реализации этой функциональности вам понадобится `action='append'`.

```
$ python3 task*_inverted_index_cli.py query --json-index  
/path/to/inverted.index --query <word> [<word> ...] --query <word>  
[<word> ...] ...
```

По результатам "обстрела" stdout должен содержать **только** ответы на запросы (всю остальную вспомогательную информацию пишите в stderr или в логи). Ответ на запрос - список идентификаторов документов (статей Википедии), разделенных запятыми. Пример:

- запрос `--query long query` состоит из двух слов "long" и "query";
- допустим в датасете только 3 документа 151, 13, 3998 содержат **одновременно оба** этих слова, тогда ваш ответ: "151,13,3998". Порядок предоставленных документов в ответе не важен (может быть любым). Но проверяется, что Вы нашли абсолютно все нужные документы и ничего лишнего;
- если на поисковый запрос не найдено ни одного документа, то нужно выводить пустую строку.

Рассмотрим пример:

```
$ python3 task*_inverted_index_cli.py query --json-index  
/path/to/inverted.index --query first query --query xxx --query the  
second query2
```

² Консоль будет работать в кодировке utf-8

Допустим ответ на первый запрос содержит документы 1,2 и 5, при ответе на второй запрос не найдено ни одного документа, а при ответе на третий запрос найдены документы 2,5,7 и 9. В этом случае STDOUT ответа должен выглядеть следующим образом:

1,2,5

2,5,7,9

2. Рекомендации

При решении задач старайтесь следовать следующим рекомендациям:

- следите за качеством кода и проверяйте “глупые” ошибки с помощью pylint, следите за поддерживаемостью и читаемостью кода;
- отделяйте фазу рефакторинга от фазы добавления новой функциональности.
 - фиксируем функциональность, все тесты зеленые;
 - проводим рефакторинг;
 - по окончании фазы рефакторинга снова все тесты зеленые;

Рекомендуем настроить виртуальное окружение Python с нужными версиями библиотек:

1. Если еще не установлено, то установите conda
<https://docs.conda.io/projects/conda/en/latest/user-guide/install/>
2. Настройте окружение для разработки на основе README.md курса
<https://github.com/big-data-team/python-course>
3. Скачайте необходимые датасеты для выполнения задания
<https://github.com/big-data-team/python-course#study-datasets>

3. Критерии оценивания

Балл за задачу складывается из:

- **20%** - получение help-string при вызове CLI
- **30%** - реализация функционала CLI для построения инвертированного индекса
 - запросы типа build должны обрабатывать в течение 5 минут на представленном в описании ДЗ датасете
- **40%** - реализация функционала CLI для обработки поисковых запросов
 - запросы типа query должны обрабатывать в течение 5 минут на представленном в описании ДЗ датасете
- **10%** - поддерживаемость и читаемость кода
 - в общем случае см. Clean Code и [Google Python Style Guide](#)
 - оценка качества будет проводиться автоматическим вызовом pylint:
 - `pylint task_*.py`
 - качество кода должно оцениваться выше 8.0 / 10.0
 - проверяем код Python версии 3.7 с помощью `pylint==2.5.3`
 - точная формула: $10\% \times \min([\text{lint_quality} / 8.0], 1.0)$

Discounts (скидки и другие акции):

- **100%** за плагиат в решениях (всем участникам процесса)
- **100%** за посылку решения после hard deadline
- **30%** за посылку решения в после soft deadline и до hard deadline
- **5%** за каждую дополнительную посылку в тестирующую систему (одна дополнительная посылка бесплатно):

лучший балл с 1-й попытки: 100%

лучший балл со 2-й попытки: 100%

лучший балл с 3-й попытки: 95%

лучший балл с 4-й попытки: 90%



4. Инструкция по отправке задания

Оформление задания:

- Код задания (Short name): **HW02:InvertedIndex CLI**
- Выполненное ДЗ запакуйте в архив `PY-MADE-2021-Q4_<Surname>_<Name>_HW#.zip`, пример `--PY-MADE-2021-Q4_Dral_Alexey_HW02.zip`. (Проверяйте отсутствие пробелов и невидимых символов после копирования имени отсюда.³) Если ваше решение лежит в папке `my_solution_folder`, то для создания архива `hw.zip` на Linux и Mac OS выполните команду⁴:
 - `zip -r hw.zip my_solution_folder/*`
- На Windows 7/8/10: необходимо выделить все содержимое директории `my_solution_folder/` нажать правую кнопку мыши на одном из выделенных объектов, выбрать в открывшемся меню "Отправить >", затем "Сжатая ZIP-папка". Теперь можно переименовать архив.
- Решение задания должно содержаться в одной папке.
- Перед проверкой убедитесь, что дерево вашего архива выглядит так:
 - `| PY-MADE-2021-Q4_<Surname>_<Name>_HW02.zip`
 - `| ---- task_<Surname>_<Name>_inverted_index_cli.py`
 - При несовпадении дерева вашего архива с представленным деревом, ваше решение не будет возможным автоматически проверить, а значит, и оценить его.
- Для того, чтобы сдать задание необходимо:
 - Зарегистрироваться и залогиниться в сервисе [Everest](#)
 - Перейти на страницу приложения: [MADE Python Grader](#)
 - Выбрать вкладку Submit Job (если отображается иная).
 - Выбрать в качестве "Task" значение: **HW02:InvertedIndex CLI**⁵
 - Загрузить в качестве "Task solution" файл с решением
 - В качестве Access Token указать тот, который был выслан по почте
- **Перед отправкой задания**, оставьте, пожалуйста, отзыв о домашнем задании по ссылке: https://rebrand.ly/pymade2021q4_feedback_hw. Это позволит нам скорректировать учебную нагрузку по следующим заданиям (в зависимости от того, сколько часов уходит на решение ДЗ), а также ответить на интересующие вопросы.

³ Онлайн инструмент для проверки: <https://www.soscisurvey.de/tools/view-chars.php>

⁴ Флаг -r значит, что будет совершен рекурсивный обход по структуре директории

⁵ Сервисный ID: `python.inverted_index_cli`



Внимание: если до дедлайна остается меньше суток, и вы знаете (сами проверили или коллеги сообщили), что сдача решений сломана, обязательно сдайте свое решение, прислав нам ссылку на выполненное задание (Job) на почту с темой письма "Short name. ФИО.". Например: **"HW02:InvertedIndex CLI. Иванов Иван Иванович."** Таким образом, мы сможем увидеть какое решение у вас было до дедлайна и сможем его оценить. Пример ссылки:

- <https://everest.distcomp.org/jobs/67893456230000abc0123def>

Любые вопросы / комментарии / предложения пишите согласно [предложениям](#) на портале.

Всем удачи!

5. FAQ (часто задаваемые вопросы)

"You are not allowed to run this application", что делать?

Если Вы видите надпись "You are not allowed to run this application" во вкладке Submit Job в Everest, то на данный момент сдача закрыта (нет доступных для сдачи домашних заданий, по техническим причинам или другое). Попробуйте, пожалуйста, еще раз через некоторое время. Если Вы еще ни разу не сдавали, у коллег сдача работает, но Вы видите такое сообщение, сообщите нам об этом.

Grader показывает 0 или < 0, а отчет (Grading report) не помогает решить проблему

Ситуации:

- система оценивания показывает оценку (Grade) < 0, а отчет (Grading report) не помогает решить проблему. Пример: в случае неправильно указанного access token система вернет -401 и информацию о том, что его нужно поправить;
- система показывает 0 и в отчете (Grading report) не указано, какие тесты не пройдены. Пример: вы отправили невалидный архив (rar вместо zip), не приложили нужные файлы (или наоборот приложили лишние - временные файлы от Mac OS и т.п.), рекомендуется проверить содержимое архива в консоли:

```
unzip -l your_solution.zip
```




**BIGDATA
TEAM**

Если Вы столкнулись с какой-то из них присылайте ссылку на выполненное задание (Job) в чат курса. Пример ссылки:

<https://everest.distcomp.org/jobs/67893456230000abc0123def>



6. Дополнительные задания (не на оценку)

Расширьте функционал библиотеки и консольный интерфейс возможностью работать со стоп-словами.

```
$ python3 task*_inverted_index_cli.py build --strategy json --dataset  
/path/to/dataset --output /path/to/inverted.index  
$ python3 task*_inverted_index_cli.py build --strategy json --dataset  
/path/to/dataset --stop-words <path> --output /path/to/inverted.index
```

Датасет со стоп-словами для выполнения задания можно скачать по адресу:

- <https://github.com/big-data-team/python-course#study-datasets>
- формат данных: одно стоп-слово на строку

Пример:

```
...  
wherein  
whereupon  
wherever  
...
```

Поскольку это задание для самостоятельной работы, то поделитесь в канале группы информацией и сравнением размера дампов с учетом и без учета стоп-слов (хеш-теги: #inverted_index #stop_words).

P.S. Не забываем отслеживать уровень покрытия тестами!