

Решение должно быть оформлено в виде компилирующегося кода с соблюдением стайл гайда. Решение задачи можно присылать (предпочтительнее ссылка на GitLab и т.п.) на e-mail [zhvv117@gmail.com](mailto:zhvv117@gmail.com) или в телеграм [@zhvv117](https://t.me/zhvv117).

Решения принимаются с 00:00 09.09.2025 до 09:00 07.10.2025 по московскому времени.

**Важно!** Сдавать решение можно не чаще 1 раза в сутки!

## Задача

Необходимо написать программу, которая выполняет эффективную сериализацию/десериализацию графа:

- В режиме `./run -s -i input.tsv -o graph.bin` программа читает граф из файла `input.tsv` в описанном ниже формате и записывает файл `graph.bin` в бинарном формате (тут нет ограничений).
- В режиме `./run -d -i graph.bin -o output.tsv` программа должна считывать `graph.bin` и записывать его в том же текстовом формате в `output.tsv`.
- Получившийся `output.tsv` может отличаться от `input.tsv` только перестановкой строк и первых двух идентификаторов в строках.

Основная задача — минимизировать размер `graph.bin`, не потеряв исходные данные.

### Формат входных данных

Рассматриваются неориентированные взвешенные по рёбрам графы без кратных рёбер, с возможными петлями, без изолированных вершин. Вершины графа имеют целочисленные идентификаторы ( $id \in [0, 2^{32} - 1]$ ). В каждой строке входного файла описывается одно ребро в виде трёх разделенных по символу табуляции целых чисел – идентификаторы смежных вершин и вес ребра (именно в таком порядке). Вес каждого ребра лежит в отрезке  $[0, 2^8 - 1]$ .

Если среди описанных рёбер не упоминается некоторая вершина с идентификатором  $v$ , нужно считать, что такой вершины нет (т.е. нет изолированных вершин).

### Как будет устроена проверка

- Присланный код должен компилироваться без сторонних библиотек, не должен использовать `gzip`, `boost`, `proto`, и т.п.
- На вход в файле `input.tsv` полученной программе будет подан граф из  $N \approx 10^6$  вершин ( $\leq 160$  MB), заведомо обладающий следующими свойствами:
  1. Граф разреженный (число рёбер не превосходит  $5N$ ).
  2. Граф имеет близкое к степенному распределение степеней вершин. То есть существуют некоторые константы  $\gamma > 1$  и  $c$ , для которых доля вершин степени  $d$  аппроксимируется при большей части значений  $d$  оценкой  $\frac{c}{d^\gamma}$ .
  3. Распределение весов рёбер и идентификаторов вершин является равномерным на соответствующих отрезках, представленных выше в описании формата входных данных.
- Полученный файл `graph.bin` будет независимо десериализован той же программой в другом режиме в `output.tsv`.
- Множества строк файлов `output.tsv` и `input.tsv` должны совпадать с точностью до перестановки первых двух чисел в них. В случае, если это не так, решение не принимается.
- Числа в tsv-файле должны быть разделены символом табуляции, не должно быть лишних пробелов и символов табуляции в начале и конце строк. В случае, если формат выходного файла оказывается неверным, решение не принимается.
- Программа должна быть однопоточной. Время работы программы (сериализации + десериализации) не должно превосходить 5 минут на ЭВМ с Intel Core i7-13800H, 64 GB RAM.

## Оценивание решений

- Задание соревновательное.
- Основной критерий оценивания решений — размер бинарного файла `graph.bin` полученный в результате сериализации закрытого теста.
- Вспомогательные критерии — дата и время отправки решения, время работы программы.
- Для удобства к заданию прилагаются примеры входных файлов `small_example.tsv` и `large_example.tsv`. Примеры входных файлов и закрытый тест сгенерированы при помощи одного и того же скрипта.
- Решения будут сравниваться как с другими решениями, так и с заранее подготовленными тремя «бейзлайнами».
- После проверки решения результаты сразу же будут видны всем.
- Номинации:
  1. Топ-3 лучших решений (по размеру бинарного файла на закрытом тесте).
  2. Первое решение, побившее Baseline 0, Baseline 1 или Baseline 2 (3 номинации, сравнивается размер бинарного файла на закрытом тесте).
  3. Самое быстрое решение, которое при этом не хуже Baseline 2.