

Третий региональный хакатон от Сбера и ВоГУ по программированию

Вологда, ВоГУ, 28-29 ноября 2025 г.

Неполная версия заданий

Хакатон включает в себя несколько заданий.

Первое задание посвящено реализации алгоритма, который вам потребуется в дальнейшем. Правильность и эффективность вашего решения будет проверяться автоматически с использованием системы Codeforces. Результаты проверки будут одним из критериев в подведении итогов.

Логин, пароль и ссылка в систему будут выданы всем командам утром 29 ноября в момент начала основного тура хакатона.

Второе и последующие задания посвящены разработке приложения, в котором вы используете решение из задания 1. На защите вы должны будете **продемонстрировать работу приложения** и ответить на вопросы жюри.

Архив с дополнительными файлами, которые вам могут потребоваться, можно скачать отсюда: <https://cloud.mail.ru/public/yqj9/5NY34ovwZ>

Задание 1 (с автоматической проверкой).

Пешие прогулки. Представьте, что вы – разработчик сайта или приложения, с помощью которого туристическая компания будет организовывать пешие прогулки туристов по городу с осмотром достопримечательностей.

Работа с вашим приложением выполняется так. Сотрудник компании выбирает N туристических объектов – кандидатов на включение в маршрут (памятников, музеев и т.п.). Далее он вводит максимальную длину маршрута D , ориентируясь на возраст, пол и физическую форму туристов.

Программа должна находить такой маршрут, чтобы он имел длину не более D и при этом содержал как можно больше выбранных достопримечательностей.

Заметим, что в начальную точку маршрута туристов привозит автобус, а затем он их забирает из конечной точки. Поэтому маршруту не требуется быть замкнутым. Также в маршруте не должно быть повторяющихся объектов.

Входные данные. В первой строке записано целое число N ($1 \leq N \leq 20$) – количество объектов. Во второй строке записано целое число D ($1 \leq D \leq 25\,000$) – максимальная длина маршрута в метрах. Далее вводится матрица a размерами $N \times N$ из целых чисел, где a_{ij} – это расстояние в метрах между объектами i и j ($0 \leq a_{ij} \leq 25\,000$). Матрица симметрична относительно главной диагонали, на главной диагонали содержатся нули.

Выходные данные. Выведите сначала количество объектов, которые вошли в маршрут, а затем номера объектов в порядке следования без повторений.

Если есть несколько ответов с максимальным количеством объектов, то выведите тот, в котором длина маршрута получится наименьшей. Если и таких ответов будет несколько, тогда выведите любой из них.

Пример ввода	Пример вывода
4 200 0 100 100 300 100 0 300 200 100 300 0 50 300 200 50 0	3 1 3 4

Система оценивания. Чем больше объектов будет в вашем маршруте, тем больше баллов вы получите. Если в вашем маршруте будет максимально возможное количество объектов, то дополнительно будет учитываться длина: чем меньше, тем лучше. Максимальное количество баллов равно 50. При занесении в лист оценок жюри этот показатель делится на 10 (например, если вы набрали 45 баллов, то в лист жюри пойдёт 4.5).

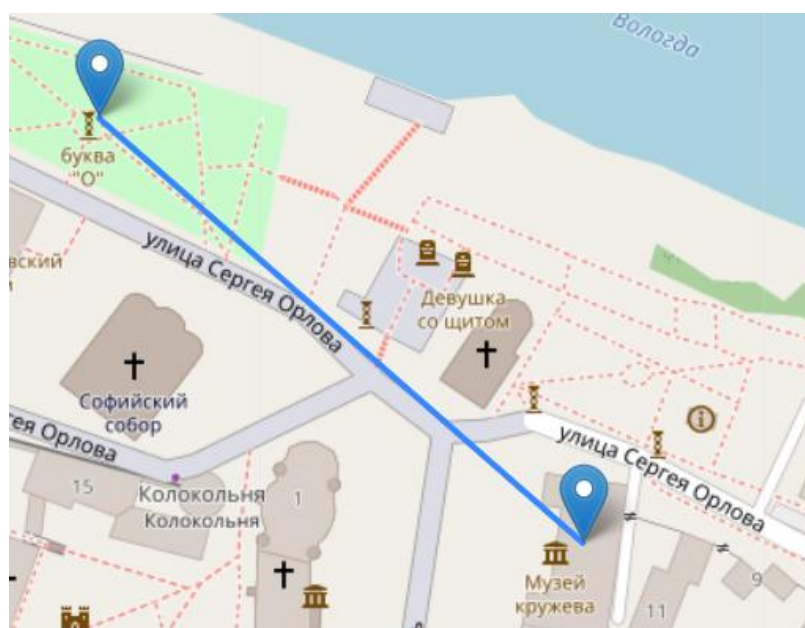
Задание 2. Разработайте Web-сайт или десктоп-приложение, о котором говорится в задании 1. Приложение будет помогать планировать пешие маршруты групп туристов по городу Вологда (команда из Череповца может при желании сделать для своего города).

В приложении должна отображаться карта города и перечень достопримечательностей (памятники, музеи). Сотрудник может выбирать интересующие его объекты двумя способами:

- отмечая элемент в перечне
- щёлкая по объекту на карте

Примечание: если вам сложно сделать оба способа, то можете ограничиться только первым (но балл при этом будет снижен).

Далее сотрудник вводит максимальную длину маршрута D и нажимает кнопку для поиска. Программа должна найти маршрут, содержащий максимальное число выбранных объектов длиной не более D , вывести объекты в порядке их посещения и нарисовать маршрут на карте. Пример, как может выглядеть маршрут от памятника букве «О» до музея кружева:



Требования и пояснения.

1. Поскольку время хакатона ограничено, для простоты будем считать, что движение между каждыми двумя объектами происходит по прямой (да и в конце концов, долго ли осталось ждать массового внедрения реактивных ранцев?)

Примечание. На практике для построения "нормальных" маршрутов по улицам/дорогам/тропам можно было бы, например, развернуть свой сервер OSRM (Open Source Routing Machine), загрузить в него карту нужной территории с OpenStreetMap и использовать API этого сервера.

2. Вы можете использовать любые инструменты, библиотеки и API, но с одним исключением: не используйте платные сервисы. Например, если вы хотите применить Яндекс- или Гугл-карты, то используйте лишь ту функциональность, которая доступна без платных подписок.

3. Возможный подход к решению заключается в использовании карт OpenStreetMap и библиотеки Leaflet. В архиве в папке javascript есть пример web-страницы, в которой отображается карта Вологды. Два туристических объекта выделены маркерами. Путь между ними изображен линией. Сделан пример обработки событий: при щелчке мышкой по любому месту карты всплывает сообщение с текущими координатами.

Аналогичный пример для Python с использованием библиотеки ipyleaflet лежит в папке python (требует установки Jupyter Notebook и библиотеки ipyleaflet).

Если вы не хотите использовать дополнительные библиотеки и API, то можете разработать приложение, используя только штатные средства выбранного языка. Для этого можно взять готовый графический файл с картой города. Пример такого файла лежит в папке img, но вы можете найти в Internet рисунок, который вам больше понравится. В своей программе показываете это изображение и поверх него рисуете выбранные туристические объекты и маршрут. При этом вам придётся найти или вывести формулы для преобразования географических координат в координаты на рисунке и наоборот.

Осталось ещё ответить на вопрос, где найти перечень туристических объектов Вологды с их координатами. Его можно получить, например, из карт OpenStreetMap. Вы можете либо самостоятельно изучить, как это сделать через API, либо применить следующий способ. В архиве в папке osm лежит файл с географическими объектами Вологды в формате XML, экспортированный из OpenStreetMap. В нём, помимо всего прочего, содержатся и достопримечательности. Например:

```
<node id="762672436" lat="59.2236047" lon="39.8815831" version="4" timestamp="2021-09-20T11:43:28Z" changeset="111445720" uid="519501" user="parukhin">
```

```
  <tag k="name" v="Краеведческий музей заповедник"/>
```

```
  <tag k="opening_hours" v="We-Su 10:00-17:30"/>
```

```
  <tag k="tourism" v="museum"/>
```

```
  <tag k="wheelchair" v="no"/>
```

```
</node>
```

Здесь атрибуты lat и lon – широта и долгота.

Ещё один пример:

```
<node id="762672602" lat="59.2229489" lon="39.8790960" version="10" timestamp="2023-08-26T23:19:29Z" changeset="140432054" uid="285369" user="avi32">
```

```
  <tag k="historic" v="memorial"/>
```

```
  <tag k="name" v="Памятник В.В. Маяковскому"/>
```

```
</node>
```

Вы можете выполнить парсинг этого файла и извлечь из него список достопримечательностей с координатами.

В субботу 29 ноября задание будет дополнено.