**Описание задачи:**

В 2035 году в Таиланде дали возможность оплачивать проезд в метро мультивалютной

криптокартой, которая поддерживает одну из трех доступных валют: **BTC, USDT, ЕТН**.

Но чтобы сэкономить, пассажиры могут оплачивать проезд из точки А в точку В в одной

валюте, а из точки **В** в точку **С** в другой.

**Цель:**

Ваше задача - разработать эндпоинт API, который вычисляет самый дешевый проезд между двумя станциями метро в Бангкоке с возможностью оплаты проезда разной криптовалютой. Стоимость проезда между станциями задана двумерными матрицами\*, каждая из которых представляет стоимость проезда в одной из криптовалют, стоимость проезда всегда положительная.

\* строки и столбцы представляют станции метро, а значения в ячейках - это стоимость проезда между этими станциями. Каждая ячейка матрицы **[i][j]** представляет собой стоимость проезда из станции **і** в

станцию **ј**

**Требования:**

1. **Используемые технологии:** РНР 8+, Laravel, Docker/sail

2. **Точность расчета:** Точность при сложениях криптовалюты должна быть 10 знаков после запятой, при операциях с USD до 2 знаков

3. **Метод АРІ:** GET запрос и название **/api/v1/cheapest-route**

4**. Параметры запроса:**

* begin - начальная станция.
* end - конечная станция.
* currencies - массив криптовалют в которых мы покупаем билет(ы)

**Начальные данные**

1. Курсы криптовалют (даны фиксированные курсы)

* USDT: 1 USDT = 1.005 USD
* ETH: 1 ETH = 3066.33 USD
* BTC: 1 BTC = 57929 USD

2. Матрицы стоимости проезда

(взять из тестового и поместить их у себя в коде,

но в учесть что в будущем эта информация могла бы запрашиваться с сайта

перевозчика, тоже самое относится к курсу валют):

**INF** в матрице означают, что прямого маршрута между этими станциями нет

Уточнение: может быть ситуация когда в одной валюте между двумя станциями можно купить билет, но в другой нельзя (расстояние между станциями INF), в этом случае строить маршрут в любой доступной валюте.

Матрицы со стоимостью маршрутов <https://pastebin.com/NXkUxFdf>

Пример запроса

GET

/api/v1/cheapest-route?begin=0&end=5&currencies[]=USDT&currencies[]=ETH&curr

encies[]=BTC

**begin** - начальная точка

**end** - конечная точка

**currencies[]** - список валют в которых мы будем пытаться купить билеты

**Пример ответа:**

code 200

{

"route": [ 0, 2, 5 ],

"total\_cost": 2.09, // стоимость проезда в USD

"tickets": [

{

"from": 0,

"to": 2,

"cost": 0.00001209,

"currency": "ВТС"

},

{

"from": 2,

"to": 5,

"cost": 0.00045656,

"currency": "ETH"

}

]

}

Если маршрута между станциями нельзя построить то АРІ должно возвращать ответ

code 422

{

"route": null,

"error": "Route from X to Y not found"

}

**Критерии оценки:**

1. Корректность работы АРІ.

2. Выбор алгоритма поиска самого дешевого маршрута.

3. Точность вычислений с использованием криптовалют.

4. Код должен быть чистым, структурированным

5. Показать знание фреймворка и его встроенных методов: DI контейнер,

сервисы, request validation, laravel resource

**Дополнительно:**

• Используйте любые библиотеки и пакеты и технологии, помощь, которые считаете нужными

• Предоставьте инструкции по запуску и тестированию вашего решения в файле readme.md, вы должны проверить ваши инструкции и убедиться что шаги описанные в них выполняются

• Установку проекта необходимо с помощью **Docker / Laravel Sail**

• Отправьте пожалуйста обратно задание в **zip/rar** архиве, это связано с тем что другие кандидаты иногда берут чужое решение из публичного репозитория

• Ориентировочное время на задание 4+ часа