Goal

Build an HTTP REST API endpoint that "routes" messages optimizing for the number of network requests.

Background Info
You work for a company (like SendHub) that sends large amounts of messages on its infrastructure. For the sake of this challenge, your job is to make sure those messages get sent using the smallest number of requests possible. For backend resources, you have at your disposal 4 different categories of message relays, each with its own rate of throughput. Here is a table of their specifications:

Category Name	relay IP address subnets	Throughput	Cost Per Request
Small	10.0.1.0/24	1 msgs/request	$0.01
Medium	10.0.2.0/24	5 msgs/request	$0.05
Large	10.0.3.0/24	10 msgs/request	$0.10
Super	10.0.4.0/24	25 msgs/request	$0.25

Requirements

1. **Functionality**
    a. API
        i. Inputs: The message and a list of valid, unique, 10-digit US phone numbers to which the message is to be sent. Note: There could be up to five thousand phone numbers in any given request.
        ii. Output: A routing of all the numbers to an appropriate IP Address within the given subnet for the desired relay category, optimizing for the constraints mentioned above. NOTE: you don't have to actually "send" any messages, just return the routing.
        iii. ***See the examples*** at the bottom of the document for example inputs and outputs.
        iv. The API should handle incorrect input gracefully, providing a useful error message to the user of the API.
2. **Constraints**
    a. You are not allowed to underutilize a request. For example, you should not route a list of 20 contacts to the "super" sized relay as it would be inefficient because it would waste 5 messages.

3. **Design**
    a. The REST API design should follow best practices when it comes to designating uniform resource identifier, the HTTP verbs used to interact with the API (GET/POST/PUT/DELETE etc).
    b. The representation used to interact with the API must support JSON
    c. The input must accept the following JSON schema:

```
{
	"title": "SendHub Challenge Schema",
	"type": "object",
	"properties": {
		"message": {
			"type": "string"
		},
		"recipients": {
			"type": "array",
			"minItems": 1,
			"items": {"type": "string"},
			"uniqueItems": true
		}
	},
	"required": ["message", "recipients"]
}
```

    * Examples of an implementation of this schema follow at the end of this document

    d. Error handling is important. Provide a consistent RESTful approach to your error handling techniques.
4. **Algorithm**
    a. Complexity
        i. Provide an analysis of the computational complexity of the algorithm you used to solve the problem.
        ii. What is its complexity?
        iii. Can you categorize this problem into the same category of other well known problems?
        iv. Is it possible to optimally solve this problem in polynomial time? What about with other throughput values?
5. **Language and Tools**
    a. Please provide your solution written in Python. You are free to use any tools or libraries that you please.
6. **Nuts and Bolts**
    a. Testing
        i. Automated Testing is not only a good way to develop, but also an easy way to prove that your code works as expected.

b. Hosting

i. We suggest using a free Tier on AWS or Heroku to host your code. Let us know if you have any issues with this or if you have already used up your free account limit. You are free to host this elsewhere but you must provide a working API for testing.

Delivery

---

1. Please get as far along as you can in the allotted time. If there are things you did not get to or further improvements you would make, please describe and prioritize them with a rationalization for their rank in priority.
2. Code and a working demo link to your server should be delivered by email.
3. You must provide sufficient documentation that the reviewer of your submission can test it.
4. You must provide working tests to prove that your code works. These tests should be in the form of cURL scripts that demonstrate the use cases your application handles.
5. Include a recent resume along with your submission
6. Include a description of your ‘biggest win’ at your most recent position
7. **IMPORTANT:** Your complete submission must be in one email. Please do not send your submission in multiple emails.

Examples:

Input:

```
{
      "message": "SendHub Rocks",
      "recipients": ["+15555555556", "+15555555555", "+15555555554", "+15555555553",
"+15555555552", "+15555555551"]
}
```

```
* Note: the above example does not include valid US phone numbers and as such should be rejected.
```

Possible Output:

```
{
      "message": "SendHub Rocks",
      "routes": [
            {
                  "ip": "10.0.1.1",
                  "recipients": ["+15555555556"]
            },
            {
                  "ip": "10.0.2.1",
                  "recipients": ["+15555555555", "+15555555554", "+15555555553",
            "+15555555552", "+15555555551"]
            },
      ]
}
* Note: the above example does not include valid US phone numbers and as such should have been rejected.
```