**Dockerized Flask App**

You are provided with the parent-child relation of the nodes as below.

Example Relation: [{"parent": "node\_1", "child": "node\_4"},

{"parent": "node\_2", "child": "node\_4"},

{"parent": "node\_3", "child": "node\_4"},

{"parent": "node\_4", "child": "node\_5"},

{"parent": "node\_4", "child": "node\_7"},

{"parent": "node\_6", "child": "node\_5"}]

This relation can be seen visually as below:

3

6

5

4

2

1111N1

7

Your task is to create a Flask app and a function with POST method which takes the relation and the node\_ids as body and return the list of nodes as Response such that each node is added to the list before their child nodes.

(Example: for node\_5 the list will be in order ["node\_1", "node\_2", "node\_3", "node\_4", "node\_6", "node\_5"])

This Flask app should be containerized using docker so you have to write a Dockerfile from which we can build a docker image and start using that flask app using the HTTP end point.

HTTP Request body example:

{"relation": [{"parent": "node\_1", "child": "node\_4"},

{"parent": "node\_2", "child": "node\_4"},

{"parent": "node\_3", "child": "node\_4"},

{"parent": "node\_4", "child": "node\_5"},

{"parent": "node\_4", "child": "node\_7"},

{"parent": "node\_6", "child": "node\_5"}],

"node\_ids": ["node\_5", "node\_7", "node\_4"]}

Expected Response:

{"node\_5": ["node\_1", "node\_2", "node\_3", "node\_4", "node\_6", "node\_5"],

"node\_7": ["node\_1", "node\_2", "node\_3", "node\_4", "node\_7"],

"node\_4": ["node\_1", "node\_2", "node\_3", "node\_4"]

}