You are required to develop a routing algorithm for the following scenarios:

1. Traffic Engineering (ability to route traffic in different paths to make sure all links of a network are used as much as possible)

2. Quality of Service (ensure the best quality of experience for all the users paying attention to the application they use)

The legacy routing protocols take only the destination IP as the input in the determination of the routing path. This, sometimes, lead to congestion in the shortest path and does not distribute the traffic through all the links. Keeping in mind the network owner (operator) makes a large investment to lay a link between two nodes and making sure that all links are used as much as possible is one way to ensure proper Return on Investment (ROI).

Let's take Traffic Engineering as an example: This could be a customer requirement or service provider requirement.

You want to route traffic from A -> B through the path that you want as opposed to the shortest path from A to B.

Understand what the service provider needs to do in order to do this. Also find out what information need to be extracted from the data plane to carry out that task. Data plane can provide you lot of information such as link status, interface bandwidth and current traffic level, packet losses if any in each link, data plane CPU and memory utilization etc., Controller can decide at what frequency it requires this information. Some information might need to be sampled at higher rate and some at lower rate.

Remember: in SDN, everything is controlled by the controller. Now develop a sequence of steps or an algorithm to carry out the given task. You need to consider all the possibilities to make sure that traffic congestion will not arise in any of the links.

Controller needs to push the right forwarding entries to the data plane to carry out the task. By pushing the entry to each data plane devices for a particular flow ID given by the five tuples, it can control traffic through the network in anyway that it likes (as per algorithm that gets applied to that particular flow ID)

Once you developed the algorithm, consider few examples of traffic engineering and see whether the algorithm that you have developed can handle that example. If Yes, then that is good. If not, refine the algorithm to do the task properly.

Algorithm needs to be developed for all the above scenarios.

Please make sure to consider end to end activity meaning from the time packet enters the network until it reaches the destination.

Please don't be constrained by legacy network limitations. You are free to use any techniques or combine multiple techniques to achieve the end outcome.

Do keep in mind security and isolation requirements.

Follow a similar process to develop a routing algorithm to assure QoS depending on the network application in use. E.g. A download, video streaming session, Voice over IP (real time voice call), a Video Conferencing session, typical web browsing all require different treatments which legacy routing protocols are unable to do.

Your algorithm needs to make sure that regardless of the network congestion, a user has the best quality of experience in accessing the end service he/she wants.