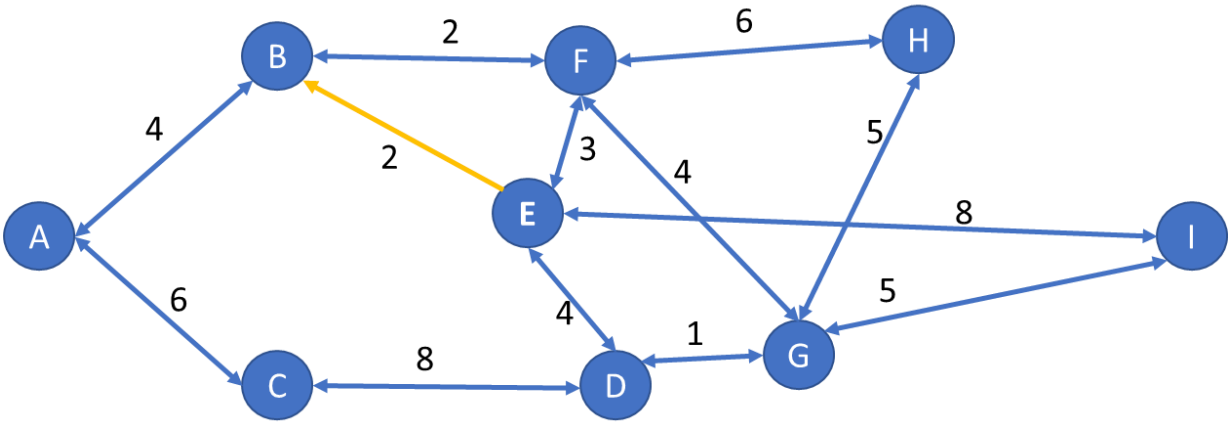
TransVirtual

Shortest Route Optimizer Testing

Design an End-to-End integration test suite to verify functional behaviour of the following sample application:

https://curious-halva-9294ed.netlify.app/

Dijkstra Shortest Path Calculator
Calculates the shortest path from one location to another location within connected nodes.



Requirements:

1. Create a manual test plan covering the following functions:
    a. Random Mode selector.
    b. Refreshing the random nodes.
    c. Selection of From/To nodes.
    d. Calculating the path.
    e. Display of visual results.
    f. Clearing the current selection.
    g. Adherence to the V1 UI design specification.
    h. Viewport scaling responsiveness of UI.
    i. API echo response OK.
    j. Browser console output of result.
2. Create an automated E2E test that executes a test runner against a browser (can be headless), covering the same scenarios created in the manual test plan.
3. Compile the results in a report of each run into a report that can be actioned by a Product Manager/Developer (depending on the issue type).

Application Business Logic:

1. Operate with 2 input modes:
    a. Take input from the user to select the FROM and TO nodes.
    b. Generate random FROM/TO node selections using an external API.
    Eg: convert a random number to a letter of the alphabet:
    http://2g.be/twitch/randomnumber.php?=defstart=1&defend=26
2. Here is the mock-up for the user input mode:
    https://www.figma.com/design/VOlrHyAO7hscTdhoUdQG6o/Coding-Challenge?node-id=0-1
3. Calculate the list of traversed nodes between FROM and TO using the above graph.
4. The node list should be in the order they are traversed.

5. Calculate the aggregate distance travelled.
6. Not all nodes are bidirectional. Eg: B can't directly go to E; but E can directly go to B.
7. The result should be captured as a DTO and output to the browser console at a regular message level:

```
type ShortestPathData = {
        readonly nodeNames: string[];
        readonly distance: number;
}
```

8. Display the results to the user. The node list should be comma separated. Eg:

```
> From Node Name = "A", To Node Name = "D": A, B, C, D
> Total Distance: 10
```

9. Send the result DTO to an HTTP echoing API and log the HTTP response to the browser console. Eg https://beeceptor.com/resources/http-echo/