Following is a walkthrough from class:

Graphical user interface, application, Teams

Description automatically generated

Recall we had a distance array and a parent array that we initialize to all 'infinity' and all blank to start. For infinity, we'll use a maximum value.

distance array:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| **I** | **I** | **I** | **I** | **I** | **I** | **I** | **I** | **I** |

parent

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|  |  |  |  |  |  |  |  |  |

We're starting at node 0 so set its distance to 0 and parent to -1

distance array:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| **0** | **I** | **I** | **I** | **I** | **I** | **I** | **I** | **I** |

parent

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| -1 |  |  |  |  |  |  |  |  |

Now start by getting the smallest distance in the distance array -> 0 for Vertex 0

Check adjacencies:

- we have 2: so add those to distance for 1 and 7

distance array:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| **0** | **4** | **I** | **I** | **I** | **I** | **I** | **8** | **I** |

parent

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| -1 | 0 |  |  |  |  |  | 0 |  |

What's the next smallest unprocessed distance (ie. exclude Vertex 0)

4 at vertex 1

Now check what's adjacent and add it to 4

- we have a distance u (4), an adjacent distance(s) (8 and 11 in this case) and potentially a distance v which is the next destination we're checking. distance v might still be I or it might have a value we can check for shortest with dist(u) + adj[u,v] and see if thats less than dist(v).

We're at vertex 1 so two adjacencies: 2 and 7 w distances 12 for 2 and 15 for 7; we won't set these as processed yet though

oops... 7 already has a distance of 8 which is less so leave that alone.

distance array:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| **0** | **4** | **12** | **I** | **I** | **I** | **I** | **8** | **I** |

parent

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| -1 | 0 | 1 |  |  |  |  | 0 |  |

So far vertices 0 and 1 are processed so now lets get the remaining smallest distance

Thats vertex 7 with distance 8 so now work from there:

We have an adjacency to 1 with total 19...not smallest so ignore that. Adjacency to 6 with total 9 < I so that can be updated and total 15 to node 8 also less than I so update that as well:

distance array:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| **0** | **4** | **12** | **I** | **I** | **I** | **9** | **8** | **15** |

parent

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| -1 | 0 | 1 |  |  |  | 7 | 0 | 7 |

0, 1 and 7 are processed so ignore them and get the smallest dist from distance

That's 9 at vertex 6. Note that every time we find the smallest value in distance to start, that's when we mark that vertex as processed.

From there we have new paths to 8 and 5 with totals of 15 and 11 respectively so ignore 8 but update 5

distance array:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| **0** | **4** | **12** | **I** | **I** | **11** | **9** | **8** | **15** |

parent

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| -1 | 0 | 1 |  |  | 6 | 7 | 0 | 7 |

Processed: { 0, 1, 7, 6 }

Next smallest is vertex 5 with distance 11 that will give us 3 totals: 15 to 2, 25 to 3 and 21 to 4. we can ignore the 15 for vertex 2 because we already have 12 for that one but update the other two:

distance array:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| **0** | **4** | **12** | **25** | **21** | **11** | **9** | **8** | **15** |

parent

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| -1 | 0 | 1 | 5 | 5 | 6 | 7 | 0 | 7 |

Processed: { 0, 1, 7, 6, 5 }

Next smallest value: vertex 2 with 12

We have three new distances: 14 to 8, 19 to 3 and 16 to 5.  we'll replace the value for 8 and 3 but leave 5 since we already have 11 for that one, so:

distance array:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| **0** | **4** | **12** | **19** | **21** | **11** | **9** | **8** | **14** |

parent

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| -1 | 0 | 1 | 2 | 5 | 6 | 7 | 0 | 2 |

Processed: { 0, 1, 7, 6, 5, 2 }

Next smallest is vertex 8 with 14. we have two distances to consider there but neither will be smaller than what we already have for 6 and 7 so there's nothing to update here.

Processed: { 0, 1, 7, 6, 5, 2, 8 }

Next smallest is 19 at vertex 3. the only unprocessed left is 4 and we already have a smaller value for that so no update.

We're now done since we count through the total number of vertices -1 and that should make sense since there are no unprocessed vertices to compare to now so we're done.

distance array:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| **0** | **4** | **12** | **19** | **21** | **11** | **9** | **8** | **14** |

parent

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| -1 | 0 | 1 | 2 | 5 | 6 | 7 | 0 | 2 |