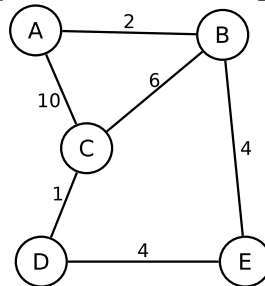


# Homework 4

*Due: 15 October 2018*

## Problem 4.1

Consider the weighted graph shown in this diagram:



Trace what Dijkstra’s algorithm does to find the shortest path from A to D. What you hand in should clearly show what got stored in the agenda, when it all arrived there, and when it got removed; but the exact format is up to you.

## Problem 4.2

In class we made a version of Dijkstra’s algorithm that returns just the *length* of the shortest path (as an `int`), but at some point we might want to fix it to return the path itself (as a `vector<string>`, as we mentioned).

Assume we use a version of `dijkstra_item` that holds “where we came from” information as a string label called `prev`. Knowing that when we find the target, we’ll have a `cur` value that is a `dijkstra_item` containing the label of that target node, its distance from the start node, and the label of the node immediately previous to it on the shortest path: how will the type of `explored` change in order to be able reconstruct the entire actual path? And what additional work will we have to do in order to actually perform the reconstruction?

(Note: as with the second Huffman problem from the first homework, this is essentially the early design phase of an implementation, but that’s all—no code is expected here!)

**Collaboration policy:** group work! If you work with other people on this homework, you can just hand in one copy and put all your names on top. There will be a revision cycle for this.