

5003CEM

Advanced Algorithms

ASSESSED AVANCED TASK
2/3

This Week's Task

This week, **there is one advanced assessed task (2/3)**. Between Weeks 4 and 9, over 6 weeks, you will be set 5 standard tasks and 3 advanced tasks, which need to be submitted, along with a report, on 9 April 2020 at 18:00.

Here are the **assessed tasks so far**:

Week 3	Assessed standard task 1	implement selection sort
Week 4	Assessed standard task 2	implement BST search
	Assessed advanced task 1	implement BST remove node method
Week 5	Assessed standard task 3	implement graph as adjacency matrix
Week 6	Assessed standard task 4	implement Prim's algorithm
	Assessed standard task 5	implement insert method for linked list (see below)
Week 7	Assessed advanced task 2	implement Dijkstra's algorithm (see below)

Here is what is yet to come:

Week 8 Assessed advanced task 3 to be announced

It is an excellent idea to write up the tasks as you go and get feedback in the Lab (sync) sessions.

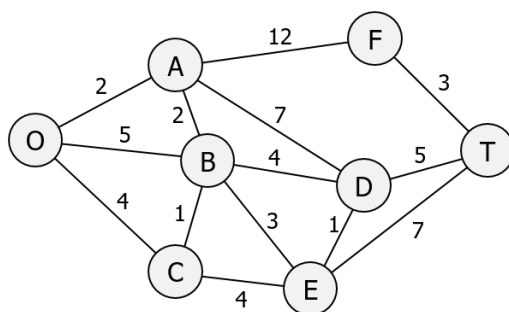
For more information, see COURSEWORK under Assessment on the Aula page.

1 Assessed Advanced Task 2: **Implement Dijkstra's Algorithm**

Advanced

Using the given code on Moodle, in the folder DIJKSTRA-CODE-ADV-TASK-2, complete the existing Python code to produce code that works for the example below (same as Week 6 lecture), plus any other (come up with alternative graphs for testing purposes). You need to complete the code where indicated.

The code, if working, will output the correct solution for the graph below.



The output from your code should look something like this – a list of the nodes which is the lowest cost path, as well as the cost itself:

```
(['O', 'A', 'B', 'D', 'T'], 13)
```