Started on	Friday, 11 June 2021, 9:12 AM
State	Finished
Completed on	Friday, 11 June 2021, 9:22 AM
Time taken	10 mins
Marks	8.00/10.00
Grade	**80.00** out of 100.00

Question **1**

Incorrect

Mark 0.00 out of 2.00

Complete the following correct statements all about Dijkstra's search.

It works on graphs with — explicit graphs only ✖

It works on — positive costs only ✖

Complexity is — O(n), where n is the number on edges ✖

Your answer is incorrect.

The correct answer is: It works on graphs with → positive costs only, It works on → both explicit and implicit graphs, Complexity is → O(n), where n is the number on nodes

Question **2**

Correct

Mark 2.00 out of 2.00

How read the shortest path from the target by reverse iteration on explicit graph?

Select one:

```
list path = {};
node = target;
while (node)
{
 path.pop(node);
 node = node.parent;
}
path.push();
path.reverse();
```

```
list path = {};
node = target;
while (node)
{
 path.push(node);
 node = node.parent;
}
path.pop();
path.reverse();
```

✔

```
list path = {};
node = target;
while (!node)
{
 path.pop(node);
 node = node.parent;
}
path.push();
path.reverse();
```

Your answer is correct.

The correct answer is:

```
list path = {};
node = target;
while (node)
{
 path.push(node);
 node = node.parent;
}
path.pop();
path.reverse();
```

Question **3**

Correct

Mark 2.00 out of 2.00

What are the data structures used in Dijkstra's search?

Closed list is implemented using Hash table ✔

Open list is implemented using Priority queue ✔

Your answer is correct.

The correct answer is: Closed list is implemented using → Hash table, Open list is implemented using → Priority queue

Question **4**

Correct

Mark 2.00 out of 2.00

Let d(u) is the total cost walking from starting node down to node u, and c(u,v) is the costs walking from u to node v.

Using drag and drop reconstruct the definition for what is called **edge relaxation** in a path searching:

For the edge from the node *u* to the node *v*, if [ ] ✔ is satisfied, update [ ] ✔ to [ ] ✔ .

d(u)+c(u,v)<d(v)

d(v)

d(u)+c(u,v)

Your answer is correct.

d(v)+c(v,u)>d(u)

d(v)+c(v,u)

r is:

d(u)

Let d(u) is the total cost walking from starting node down to node u, and c(u,v) is the costs walking from u to node v.

Using drag and drop reconstruct the definition for what is called **edge relaxation** in a path searching:

For the edge from the node *u* to the node *v*, if *[d(u)+c(u,v)<d(v)]* is satisfied, update *[d(v)]* to *[d(u)+c(u,v)]*.

Question **5**

Correct

Mark 2.00 out of 2.00

Check all correct statements about Manhattan heuristic.

Select one or more:

- Calculated as a total number of squares from starting node to target ✖
- Using a heuristic is what makes A* different from Dijkstra's search ✔
- Used in both Dijkstra's and A* searches
- Used in A*, but not in Dijkstra's search. ✔

Your answer is correct.

The correct answers are: Used in A*, but not in Dijkstra's search., Using a heuristic is what makes A* different from Dijkstra's search

◄ Attendance cs380su21-meta.sg Friday 11/6/2021 9:00am-12:20pm

Jump to...

Slides (Bellman-Ford's Search) ►