

■ All Topics > Search Results

Add a Post

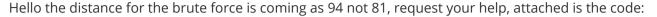
Search all posts lab 5 C

Search

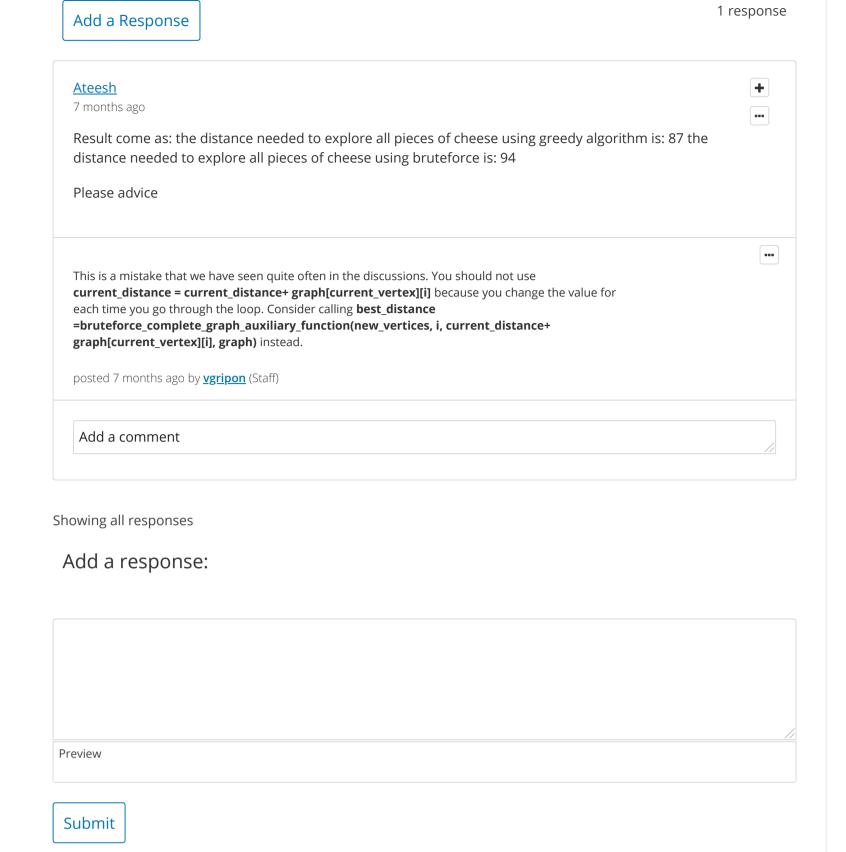
Show all posts ✓ by recent activity ✓		
?	Lab 5: Exercise C- Wrong Answer Hi, My Brute-force algorithm	5
2	Lab 5 exercise C Distance for Bruteforce Hello the distance for the br	3
?	Lab 6: Strange Results	6
?	a headache question in lab5 exercise C Hi, for the exercise C of the L	4
2	<u>Lab 5, C</u> <u>Dear all, I cannot make exer</u>	3
€	How to I read the adjacency graph? Hi, I am stuck at exercise C o	6
2	<u>Lab 5 (c)</u>	3
?	lab 5 Exercise C Where the instructions say:	21
?	lab 5 Exercise C In the description of the gre	18

Lab 5 exercise C Distance for Bruteforce

discussion posted 7 months ago by $\underline{\textbf{Ateesh}}$



```
infinity = 10000000
def greedy_complete_graph(graph, initial_vertex):
    current_distance = 0
    # while we still have a vertex that is not explored
         get next target using choose_target function
        update player position
        update distance value
        remove next target from vertices list
    vertices=list(range(len(graph)))
    vertices.remove(initial_vertex)
    current_initial_vertex = initial_vertex
    while (vertices!=[]):
        next_target=choose_target(graph[current_initial_vertex],vertices)
        current_distance+=graph[current_initial_vertex][next_target]
        current_initial_vertex=next_target
        vertices.remove(next_target)
    return(current_distance)
best_distance = infinity
def bruteforce_complete_graph_auxiliary_function(remaining_vertices, current_vertex,
current_distance, graph):
    global best_distance
    # if there are no remaining vertices:
       if the current distance is shorter than the best distance:
          update the best distance to match the current distance
    # else:
       for each vertex in the remaining vertices:
          perform a copy of the remaining vertices, let us call them "new_vertices"
          remove the vertex from new_vertices
          call the function bruteforce_complete_graph_auxiliary_function recursively
    if len(remaining_vertices)==0:
        if current_distance < best_distance:</pre>
            best_distance= current_distance
    else:
        for i in remaining_vertices:
            new_vertices = remaining_vertices.copy()
            new_vertices.remove(i)
            current_distance = current_distance+ graph[current_vertex][i]
            best_distance =bruteforce_complete_graph_auxiliary_function(new_vertices, i,
current_distance, graph)
    return best_distance
def bruteforce_complete_graph(graph, initial_vertex):
    global best_distance
    best_distance = infinity
    remaining_vertices = list(range(len(graph)))
    remaining_vertices.remove(initial_vertex)
    bruteforce_complete_graph_auxiliary_function(remaining_vertices, initial_vertex, 0, graph)
    return best_distance
```





edX

About

edX for Business

Legal

Terms of Service & Honor Code **Privacy Policy**

Accessibility Policy

Connect

Blog

Contact Us

Help Center













© 2020 edX Inc. All rights reserved.

|深圳市恒宇博科技有限公司 <u>粤ICP备17044299号-2</u>