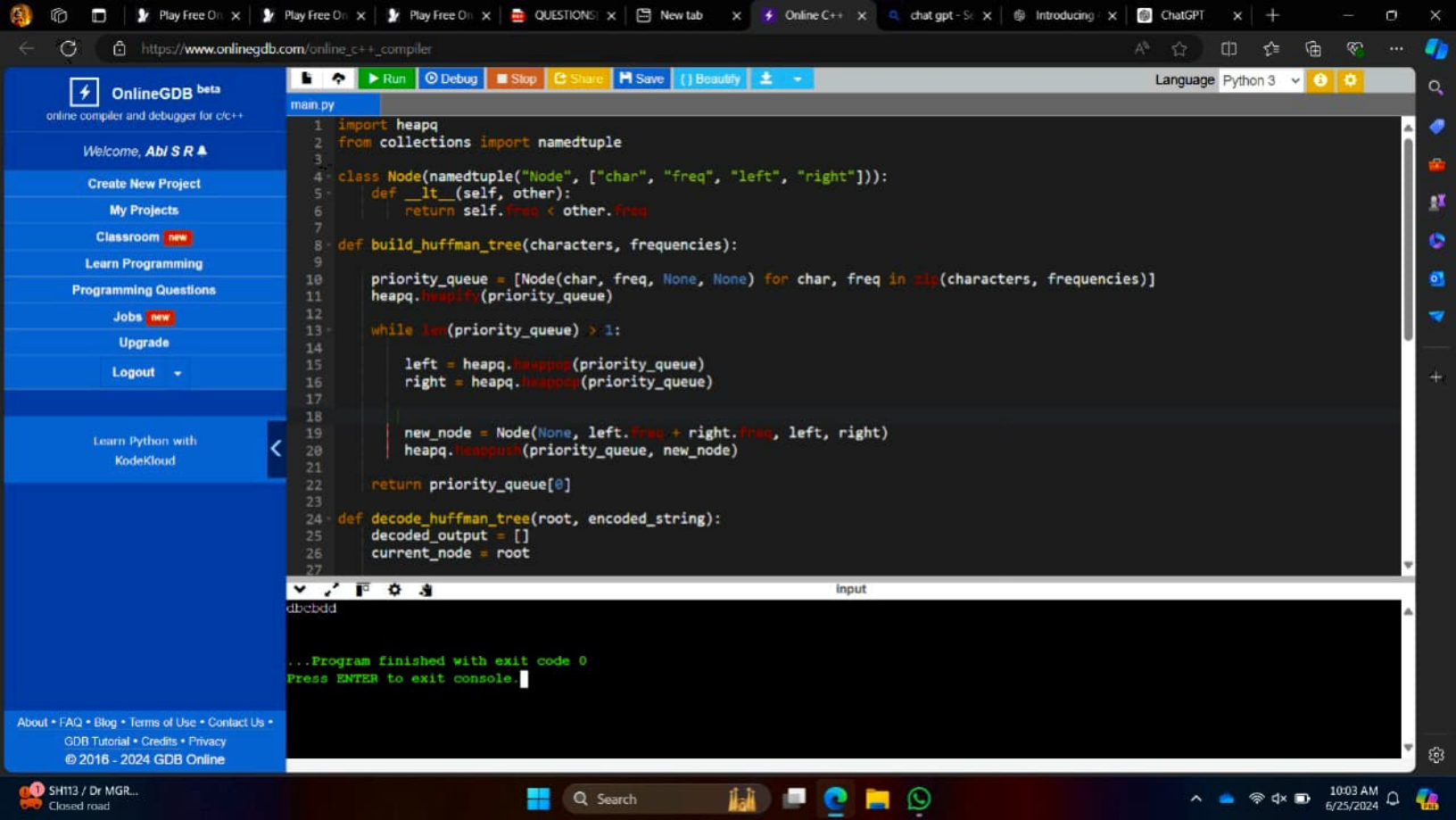
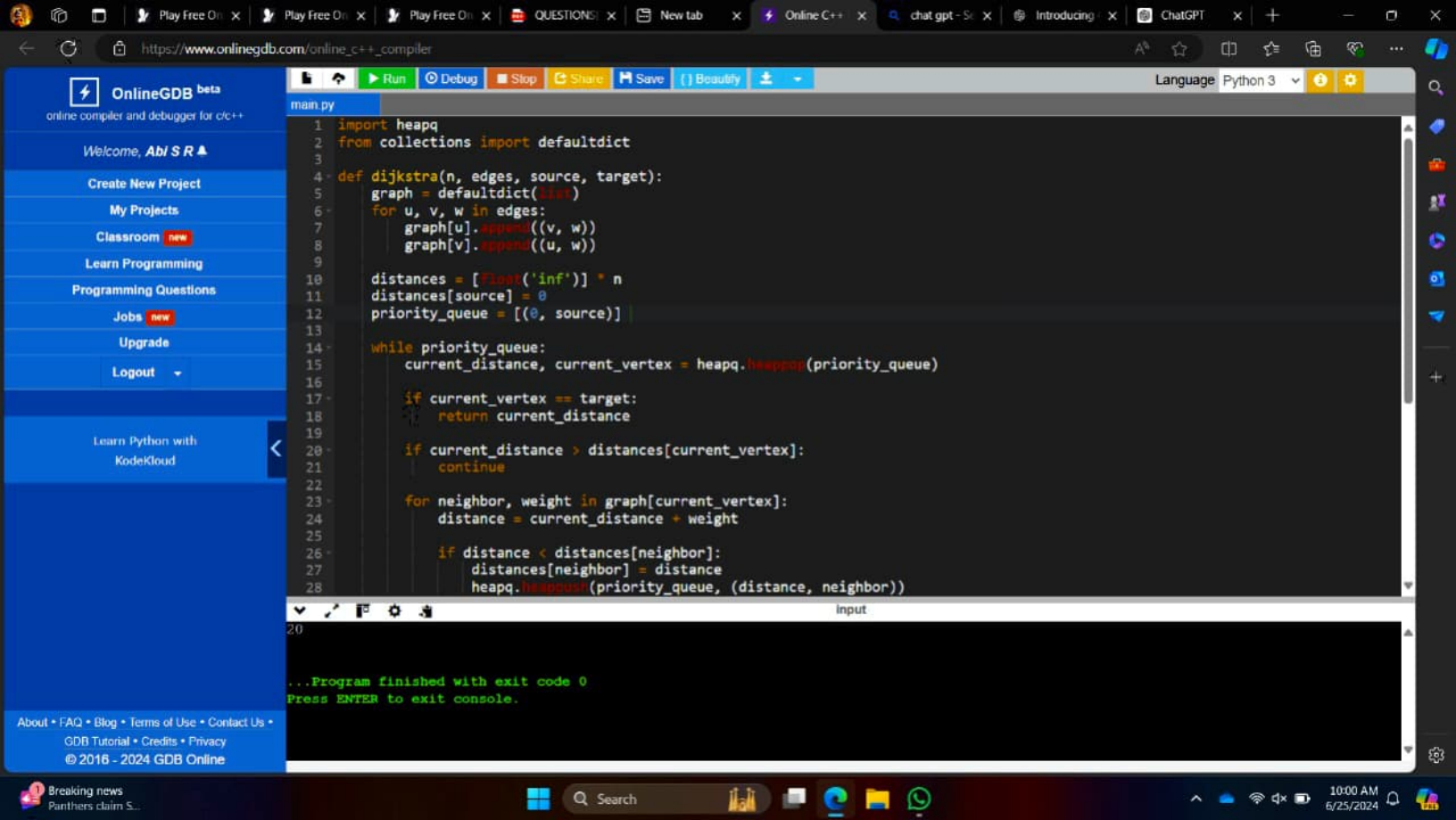
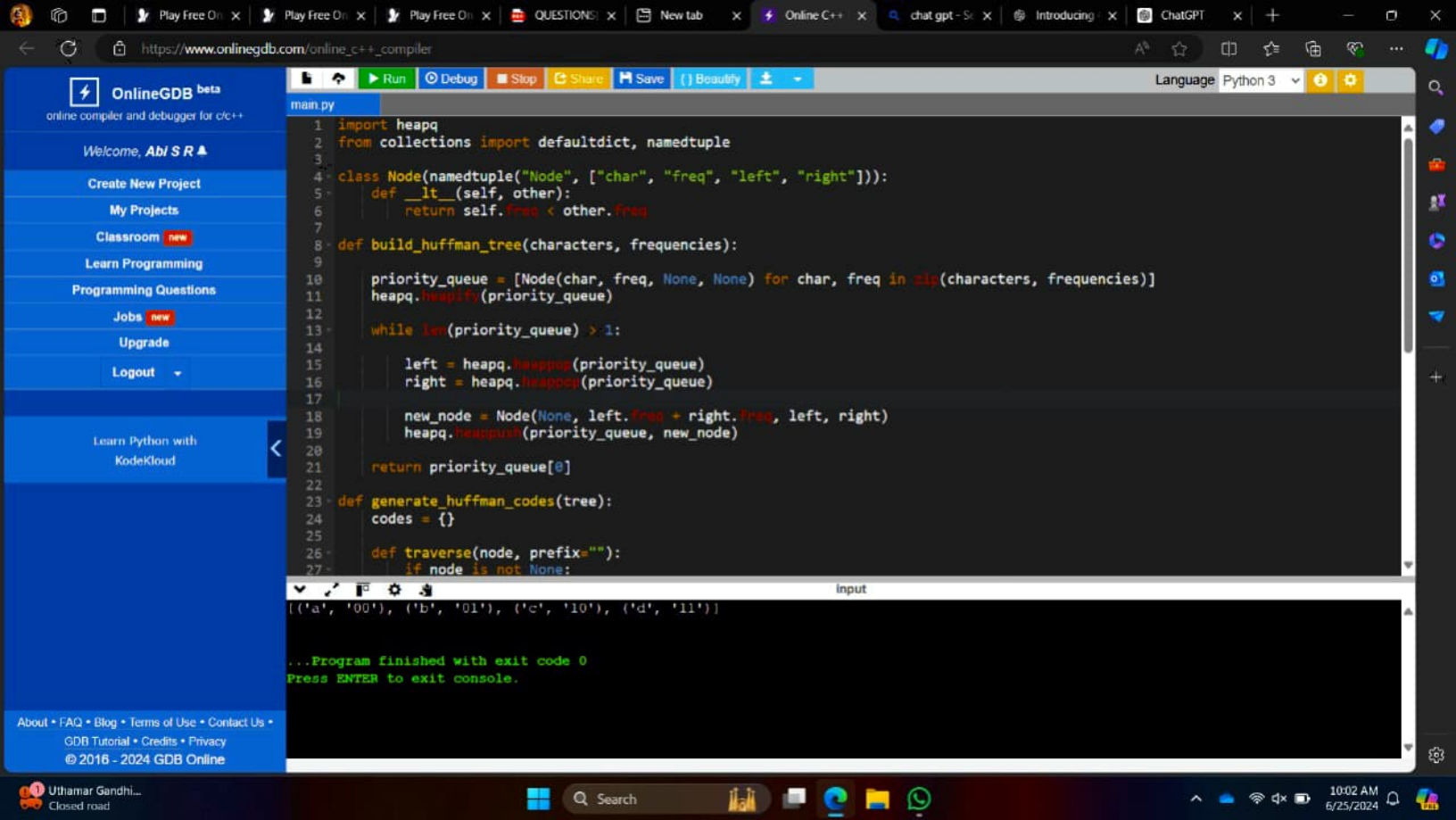
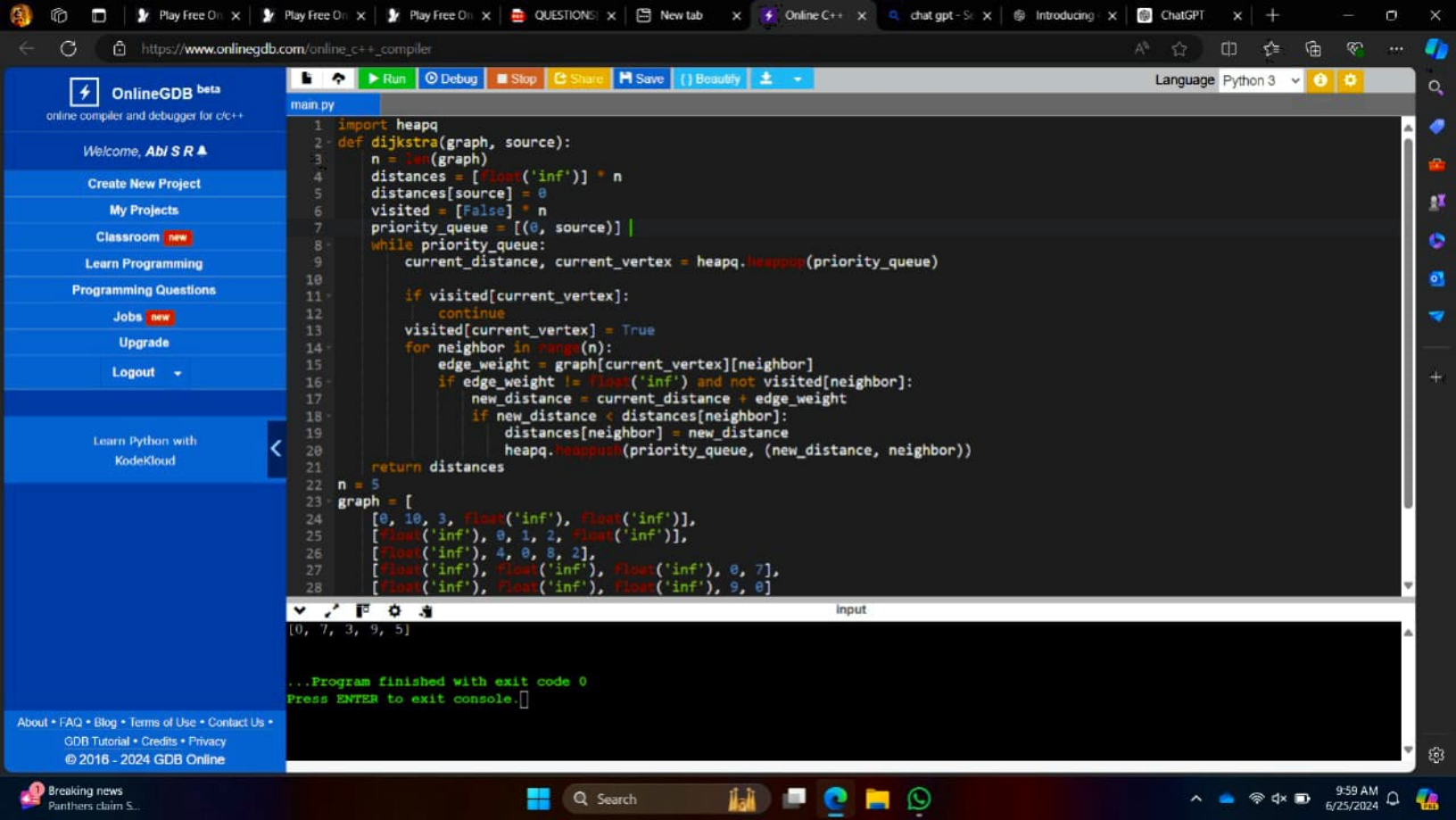

OnlineGDB beta
online compiler and debugger for c/c++
Welcome, Abi S R
Create New Project
My Projects
Classroom new
Learn Programming
Programming Questions
Jobs new
Upgrade
Logout
Learn Python with KodeKloud
Run
Debug
Stop
Share
Save
{ } Beautify
Language
Python 3
main.py
import heapq
from collections import namedtuple
class Node(namedtuple("Node", ["char", "freq", "left", "right"])):
def __lt__(self, other):
return self.freq < other.freq
def build_huffman_tree(characters, frequencies):
priority_queue = [Node(char, freq, None, None) for char, freq in zip(characters, frequencies)]
heapq.heapify(priority_queue)
while len(priority_queue) > 1:
left = heapq.heappop(priority_queue)
right = heapq.heappop(priority_queue)
new_node = Node(None, left.freq + right.freq, left, right)
heapq.heappush(priority_queue, new_node)
return priority_queue[0]
def decode_huffman_tree(root, encoded_string):
decoded_output = []
current_node = root
Input
dbcbdd
...Program finished with exit code 0
Press ENTER to exit console.
About • FAQ • Blog • Terms of Use • Contact Us •
GDB Tutorial • Credits • Privacy
© 2016 - 2024 GDB Online


Play Free On
Play Free On
Play Free On
QUESTIONS
New tab
Online C++
chat gpt - Se
Introducing
ChatGPT
https://www.onlinegdb.com/online_c++_compiler
OnlineGDB beta
online compiler and debugger for c/c++
Welcome, Abi S R
Create New Project
My Projects
Classroom new
Learn Programming
Programming Questions
Jobs new
Upgrade
Logout
Learn Python with KodeKloud
Run
Debug
Stop
Share
Save
{ } Beautify
Language
Python 3
main.py
import heapq
from collections import defaultdict
def dijkstra(n, edges, source, target):
graph = defaultdict(list)
for u, v, w in edges:
graph[u].append((v, w))
graph[v].append((u, w))
distances = [float('inf')] * n
distances[source] = 0
priority_queue = [(0, source)]
while priority_queue:
current_distance, current_vertex = heapq.heappop(priority_queue)
if current_vertex == target:
return current_distance
if current_distance > distances[current_vertex]:
continue
for neighbor, weight in graph[current_vertex]:
distance = current_distance + weight
if distance < distances[neighbor]:
distances[neighbor] = distance
heapq.heappush(priority_queue, (distance, neighbor))
Input
20
...Program finished with exit code 0
Press ENTER to exit console.
About • FAQ • Blog • Terms of Use • Contact Us •
GDB Tutorial • Credits • Privacy
© 2016 - 2024 GDB Online
Breaking news
Panthers claim S...
Search
10:00 AM
6/25/2024


OnlineGDB beta
online compiler and debugger for c/c++
Welcome, Abi S R
Create New Project
My Projects
Classroom new
Learn Programming
Programming Questions
Jobs new
Upgrade
Logout
Learn Python with KodeKloud
Run
Debug
Stop
Share
Save
{ } Beautify
Language Python 3
main.py
import heapq
from collections import defaultdict, namedtuple
class Node(namedtuple("Node", ["char", "freq", "left", "right"])):
def __lt__(self, other):
return self.freq < other.freq
def build_huffman_tree(characters, frequencies):
priority_queue = [Node(char, freq, None, None) for char, freq in zip(characters, frequencies)]
heapq.heapify(priority_queue)
while len(priority_queue) > 1:
left = heapq.heappop(priority_queue)
right = heapq.heappop(priority_queue)
new_node = Node(None, left.freq + right.freq, left, right)
heapq.heappush(priority_queue, new_node)
return priority_queue[0]
def generate_huffman_codes(tree):
codes = {}
def traverse(node, prefix=""):
if node is not None:
Input
[('a', '00'), ('b', '01'), ('c', '10'), ('d', '11')]
...Program finished with exit code 0
Press ENTER to exit console.
About • FAQ • Blog • Terms of Use • Contact Us • GDB Tutorial • Credits • Privacy
© 2016 - 2024 GDB Online
Search
10:02 AM
6/25/2024


Play Free On
Play Free On
Play Free On
QUESTIONS
New tab
Online C++
chat gpt - Se
Introducing
ChatGPT
https://www.onlinegdb.com/online_c++_compiler
OnlineGDB beta
online compiler and debugger for c/c++
Welcome, Abi S R
Create New Project
My Projects
Classroom new
Learn Programming
Programming Questions
Jobs new
Upgrade
Logout
Learn Python with KodeKloud
Run
Debug
Stop
Share
Save
{ } Beautify
Language
Python 3
main.py
import heapq
def dijkstra(graph, source):
n = len(graph)
distances = [float('inf')] * n
distances[source] = 0
visited = [False] * n
priority_queue = [(0, source)]
while priority_queue:
current_distance, current_vertex = heapq.heappop(priority_queue)
if visited[current_vertex]:
continue
visited[current_vertex] = True
for neighbor in range(n):
edge_weight = graph[current_vertex][neighbor]
if edge_weight != float('inf') and not visited[neighbor]:
new_distance = current_distance + edge_weight
if new_distance < distances[neighbor]:
distances[neighbor] = new_distance
heapq.heappush(priority_queue, (new_distance, neighbor))
return distances
n = 5
graph = [
[0, 10, 3, float('inf'), float('inf')],
[float('inf'), 0, 1, 2, float('inf')],
[float('inf'), 4, 0, 8, 2],
[float('inf'), float('inf'), float('inf'), 0, 7],
[float('inf'), float('inf'), float('inf'), 9, 0]
Input
[0, 7, 3, 9, 5]
...Program finished with exit code 0
Press ENTER to exit console.
About • FAQ • Blog • Terms of Use • Contact Us • GDB Tutorial • Credits • Privacy
© 2016 - 2024 GDB Online
Breaking news
Panthers claim S...
Search
9:59 AM
6/25/2024