

 **OnlineGDB** beta  
online compiler and debugger for c/c++

code. compile. run. debug. share.

IDE

My Projects


Classroom new



Learn Programming

Programming Questions

Sign Up

Login









   12.4K



  
**GOT AN OPINION?**  
SHARE AND GET REWARDED.  


Have fun taking surveys  
and get paid!

ADS VIA CARBON

About • FAQ • Blog • Terms of Use • Contact Us • GDB  
Tutorial • Credits • Privacy  
© 2016 - 2020 GDB Online

  Run  Debug  Stop  Share  Save  { } Beautify 

Language Python 3  

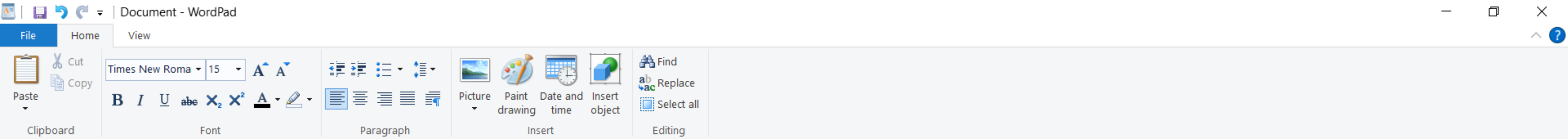
```
main.py
1 graph = {
2     'A' : ['B','C'],
3     'B' : ['D', 'E'],
4     'C' : ['F'],
5     'D' : [],
6     'E' : ['F'],
7     'F' : []
8 }
9
10 visited = set() # Set to keep track of visited nodes.
11
12 def dfs(visited, graph, node):
13     if node not in visited:
14         print (node)
15         visited.add(node)
16         for neighbour in graph[node]:
17             dfs(visited, graph, neighbour)
18
19 # Driver Code
20 dfs(visited, graph, 'A')
```

input

A  
B  
D  
E  
F  
C

Activate Windows  
Go to Settings to activate Windows.





## Depth-First Search

Depth-first search (DFS), is an algorithm for tree traversal on graph or tree data structures. It can be implemented easily using recursion and data structures like dictionaries and sets.

## The Algorithm

1. Pick any node. If it is unvisited, mark it as visited and recur on all its adjacent nodes.
2. Repeat until all the nodes are visited, or the node to be searched is found.

## Time Complexity

Since all the nodes and vertices are visited, the average time complexity for DFS on a graph is  $O(V+E)O(V+E)O(V+E)$ , where  $V$  is the number of vertices and  $E$  is the number of edges. In case of DFS on a tree, the time complexity is  $O(V)O(V)O(V)$ , where  $V$  is the number of nodes.