物件導向程式設計

Case Study IV:
Graphs Basics & DFS

Joseph Chuang-Chieh Lin
Dept. CSIE, Tamkang University

Platform

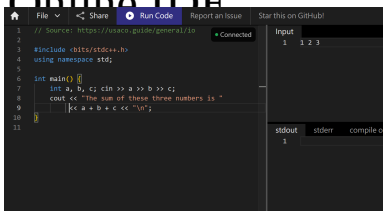
- Dev-C++

  Click here to download.

  Note: Please use this version otherwise you can't compile your programs/projects in Win10.

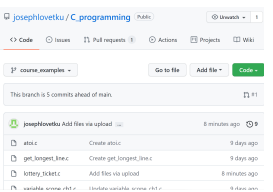
- OnlineGDB (https://www.onlinegdb.com/)

- Real-Time Collaborative Online IDE (https://ide.usaco.guide/)



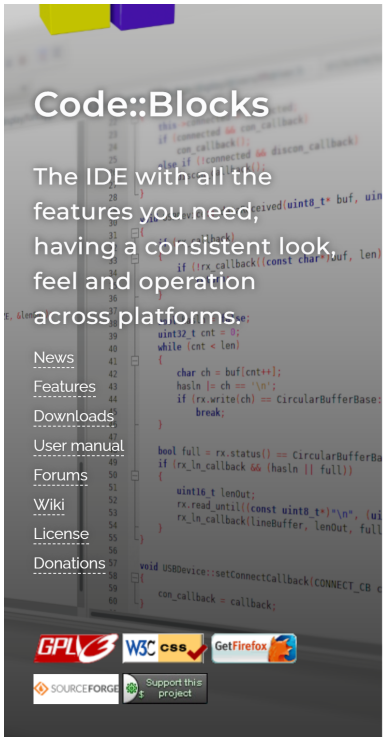
- Other resources:
  - MIT OpenCourseWare - Introduction to C++ [link].
  - Learning C++ Programming [Programiz].
  - GeeksforGeeks [link]

My GitHub page:
click the link here to visit.



Platform/IDE

- https://www.codeblocks.org/



Code::Blocks

Code::Blocks

The free C/C++ and Fortran IDE.

Code::Blocks is a free C/C++ and Fortran IDE built to meet the most demanding needs of its users. It is designed to be very extensible and fully configurable.

Built around a plugin framework, Code::Blocks can be extended with plugins. Any kind of functionality can be added by installing/coding a plugin. For instance, event compiling and debugging functionality is provided by plugins!

If you 're new here, you can read the **user manual** or visit the **Wiki** for documentation. And don't forget to visit and join our **forums** to find help or general discussion about Code::Blocks.

We hope you enjoy using Code::Blocks!

The Code::Blocks Team

Latest news

Migration successful

We are very happy to announce that the process of migrating to the new infrastructure has completed successfully!

Read more

Some basics on graphs

- https://web.ntnu.edu.tw/~algo/Graph.html

DFS

- DFS can be implemented by recursion.

```
Depth-First-Search_Init(graph m )
    Depth-First-Search(m.start_node)
End procedure

Depth-First-Search(vertex c )
    If c is the goal
        Exit
    Else
        Mark c "Visit In Progress"
        For each neighbor n of c
            If n is "Unvisited"
                Depth-First-Search(n)
        Mark c "Visited"
End procedure
```

- **Issue:** Explosion in the recursion stack

DFS using a stack

- Using STL::stack
- Reference code: https://onlinegdb.com/i1iB8hW2G

Exercise

- Use the DFS code to enumerate connected components.
- Sample input:

```
5
1 0
2 3
3 4
```

- Sample output:

```
0 1
2 3 4
```