

VSCode 相关

如果你使用 VSCode 的 Code Runner 插件提供的 Run Code 按钮（快捷键默认为 `ctrl+alt+n`），Code Runner 会根据你的文件类型来执行一条预先设定好的指令。以 C++ 代码文件为例，Code Runner 执行的指令是位于当前打开的文件夹下的 `.vscode\settings.json` 中的 `"code-runner.executorMap"` 的 `"cpp"` 项的值（如果当前 workspace 下的 `.vscode\settings.json` 中没有该项，则取全局设置中的值）。如果你作如下设置（注意在 JSON 的字符串中 `\` 需要打两个）

```
{
  "code-runner.executorMap": {
    "cpp": "cd $dir && g++ '$fileName' path\\to\\your\\graph.cpp -o '$fileNameWithoutExt' -std=c++20 -Wall -Wpedantic -Wextra && echo 'compilation ends.' && .\\'$fileNameWithoutExt'"
  }
}
```

则每次编译一个文件都会带上 `graph.cpp`。这里你需要将 `path\to\your\graph.cpp` 替换为 `graph.cpp` 所在的位置，你可以直接写一个绝对路径，如 `D:\cs101\pa3\attachments\graph.cpp`。

如果你使用 C/C++ 插件的 Run C/C++ File 或者 Debug C/C++ File（即调试）功能，它对应的配置是当前打开的文件夹下的 `.vscode\tasks.json`。它的内容类似于

```
{
  "tasks": [
    {
      "type": "cppbuild",
      "label": "C/C++: g++ build active file",
      "command": "path\\to\\your\\g++.exe",
      "args": [
        "-fdiagnostics-color=always",
        "-g",
        "${file}",
        "path\\to\\your\\graph.cpp",
        "-o",
        "${fileDirname}/${fileBasenameNoExtension}",
        "-Wall",
        "-Wpedantic",
        "-Wextra",
        "-std=c++20",
      ],
      "options": {
        "cwd": "${fileDirname}"
      },
      "problemMatcher": [
        "$gcc"
      ],
      "group": {
        "kind": "build",
        "isDefault": true
      },
      "detail": "Task generated by Debugger."
    }
  ]
}
```

```
],  
  "version": "2.0.0"  
}
```

将上面的 `path\to\your\g++.exe` 和 `path\to\your\graph.cpp` 分别改成你的 `g++.exe` 和 `graph.cpp` 所在的路径（可以直接写绝对路径），启动 Run C/C++ File 或者 Debug C/C++ File 的时候就会自动带上 `graph.cpp` 了。