

Review Form

Authors's Name: 吴凭飞 Author's Student No.: 2018011366

Reviewer's Name: 吕时清 Reviewer's Student No.: 2018013379 Date: 05-19

Makefile (10%)	10
Review Comments	The one supplied in the requirement has its deficiencies, but your code works with it well. Nicely done.
Compilation (编译正确) (10%)	9
Review Comment	A tiny issue — BaseRouter.cxx is missing in problem1/ version b. Everything else compiles correctly.
Correctness of Results (结果正确) (40%)	40
Review Comment	You probably would like to ensure that $\text{newX} \geq 0$ in OptRouter.cxx, line 39. That apart, the searching algorithm is well-implemented. The code for problem 2 works correctly.
Naming Convention (变量命名合理) (5%)	5
Review Comment	Short and concise within every context.
Code Formatting (代码格式合理) (5%)	4
Review Comment	The mixed usage of spaces and tabs is a minor concern, and inconsistent indentation arises occasionally. Fair otherwise.
Code Comments (代码注释合格) (5%)	4
Review Comment	Maybe add some to clarify the work of functions and meaning of variables. e.g. What member variables does countturns() operate on? What do x and y mean for searchroute()? What is the contents of <allpaths>?
Other Coding Style and efficiency (代码运行效率) (10%)	10
Review Comment	There is a decent space for optimization in the searching algorithm, but it's already great for our problem!
OOP Design Style (15%)	15

Review Comment	Good reuse of provided code. Though I'd suggest OptRouter::searchroute() and similar members be made protected or private.
Total Score (0-100)	97
Overall Review Comments	The search algorithm is well-implemented, and the bug in problem 2 correctly resolved. Kudos! On a side note, maybe when one wishes to share the code to others, the latter is expected to at least do a little bit of review ^ ^
Version of Reviewer's compiler	Apple LLVM version 9.1.0 (clang-902.0.39.2) GNU Make 3.81

NOTE:

1. For coding styles including naming, formatting, code comments, etc., please refer to the Google C++ Style Guide. You may stick to your own coding style if you already have one, provided that your coding style is easy to read and understand by others.
(<https://google.github.io/styleguide/cppguide.html>)
2. OOP design: easy for code reuse (代码复用); easy for extension and adaptability to future change for new user requirements (易于扩展, 适应未来用户需求的改变). We will learn these OOP design features throughout this semester.
3. The reviewer (评阅人) is responsible for filling in the review form with both credits (分数) and detailed comments/suggestions (评价/建议) for further improvements (改进), and then returns the review form to the author of the code.
4. Finally, each student needs to submit the improved (改进后的) source code according to the comments, as well as a .doc or .pdf file including the returned review forms and statements on the revised code in reply to the reviewers' comments.
5. Please comply with “**Rules for Submission** (see Lecture 1: Exercises)” when submitting your homework.