

# Programming Project 3

Due **12/6/11** before class

## Prim's MST algorithm (20 points)

Implement two variants of Prim's MST algorithm: One that stores the weighted graph in an adjacency matrix, and the other that stores the weighted graph in adjacency lists. (Do not simply convert one representation into the other, but actually implement your algorithms to directly work with the specified input representation of the graph.) You can either use a given implementation of a priority queue, or you can replace the efficient priority queue implementation by a (slower) workaround that uses simpler array operations.

Test your algorithm on the graph shown on slide three of the MST slides. The output of your algorithm should include the predecessor array storing the MST.

Your code and project directory should contain the following:

- A way to input the graph. (Hard-coding is fine if documented properly.)
- A way to output the resulting MST (at least using the predecessor array).
- A short test-case report that documents how your code runs on the graph given on slide three of the MST slides.

## Turnin instructions

- You are allowed to turn in this programming project in groups of two.
- You can use Java, C, or C++ for this project. If you want to use a different programming language, check with our TA first.
- **The name of your project directory should be** `project3_<language><lastName1><firstName1><lastName2><firstName2>`, where `<language>` is the programming language you are using.
- Zip up a directory with your entire project (source code and test case report). Turn in the zip file by uploading it to Blackboard. In the comments section during the upload to Blackboard please add instructions on how to compile the program and on how to run the test cases.
- All projects need to compile. If your program does not compile you will receive 0 points on this project.
- Do not use any fancy libraries. We should be able to compile it under standard installs of Java, C, or C++ under linux and/or windows.