Name:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Directions To Evaluator:** Please fill in each of the blank spaces (either during the presentation, or afterwards) based on what is presented by your peer. This sheet can also be used as a study-guide for yourself, later on.

1. **Introduction / Identification of the algorithm to be presented**The problem that the algorithm solves:  
     
   Explain how it compares to other algorithms, in a couple quick sentences. In a nutshell, why would anybody want to want to use it?
2. **Properties of the algorithm, notes on the algorithm**
   1. Running time  
       Min: Avg: Max:
   2. Space required  
        
        
      Limitations and Strengths  
      For a sorting algorithm, can it sort an array? A linked list?   
        
      Are there any particular situations wherein the algorithm does badly / can't operate at all, or where it does well?
3. **Evaluator Comments:** (Your summary goes here: )

Evaluation Rubric

Each presentation will be graded, in part, with the help of the following rubric. Each category is rated on a scale of 1 to 5, 1 being bad and 5 being good.

|  |  |
| --- | --- |
| **Technical Knowledge How well you convey a sound, technical understanding of the material.** | **Score** |
| 1. Able to provide a sound motivation for using this particular algorithm.  Including concrete examples of use. |  |
|  | |
| 1. Sound understanding of the algorithm and it’s properties |  |
|  | |
| * + Both overall, and details |  |
|  | |
| 1. Correct description (including pseudo-code) |  |
|  | |
| 1. Correct analysis (running time, etc) |  |
|  | |
| 1. Code correctly implemented program works (in C++, C#, Java, etc) |  |
|  | |
| **Preparation The work you do prior to your presentation** | |
| 1. Prepared in advance |  |
|  | |
| 1. Either no technical problems during presentation, or else they’re dealt with quickly and efficiently. Unforeseen problems that can't be solved during the presentation should be worked around in a professional manner. |  |
|  | |
| 1. Makes effective use of available resources. You're not required to use anything in particular, but merely standing in front of the class and talking isn't acceptable. You should produce several concrete objects (such as PowerPoint slides, notes, handouts, etc) |  |
|  | |
| 1. Lasts about 20-30 minutes |  |
|  | |
| 1. Citations (Were the sources that you used clearly cited?) |  |
|  | |
| **Delivery The actual execution; How the materials are delivered to the audience.** | |
| 1. Presentation begins promptly. If you need to log in, set up MSVC++, or load a web page, then you've done this prior to the start of class. Overhead projector is warmed up. |  |
|  | |
| 1. Explanation is clear, including verbal description, whiteboard run-throughs, and C++ source code |  |
|  | |
| 1. Explanation is concise. |  |
|  | |
| 1. Questions are addressed in a clear and effective manner. |  |
|  | |
| 1. Every member of the group does a roughly equal amount of work during the actual presentation. |  |
|  | |

**Grade:**   
(out of 100)