Meaningful Title for this Paper

John Doe

Abstract

The abstract is written *after* the report and gives a high–level overview. An abstract is optional for these reports, but it is good practice. Look up some published papers and read the abstracts to get an idea of how to write one.

I. Motivation

There has to be some good reasons to do this work and publish it. Why? Never refer to any report for this class as a class project, lab project, or assignment. It is a report, a work, or an article. Try to be professional.

II. Background

What exactly is a good description of the problem? What work has been done on it? In the past, others have reported algorithms to perform an Insertion Sort [**?**]. Remember: If you copy it, it's plagiarism until you cite it.

III. Procedure

How did you do this wonderful thing anyway? This is where you put your pseudocode (see Algorithm 1), pre-conditions, post conditions, and Invariants. Describe the idea in a general way. How you implement your procedure goes in the Experimental Analysis.

Algorithm 1 Build-Max-Heap(A)

```
1: procedure Build-Max-Heap(A)
2:     assert(A is not empty)
3:     heapsize = A.length
4:     for i = ⌊heapsize/2⌋ downto 2 do
5:         Max-Heapify(A,i)
6:         assert(Heap-Test(A,i)
7:     end for
8:     assert(Heap-Test(A,1)
9: end procedure
```

IV. Testing

A. Testing Plan and Results

This can be done as a table. Something like:

Test input	Expected Results	Actual Results	Corrective Actions
Empty array	nothing returned	nothing was returned	no action taken
All elements identical	same array	Program faulted	off–by–one error corrected

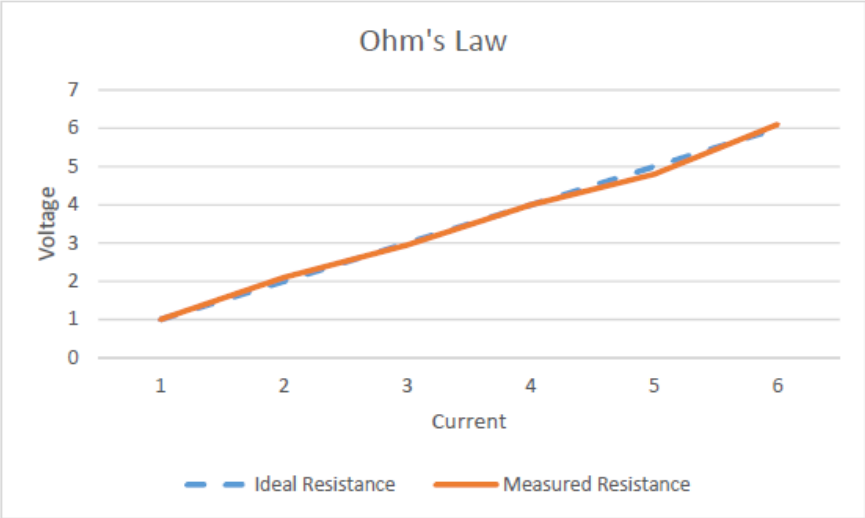

Figure 1. A Sample Graph

B. Problems Encountered

Normally, you would not see these in a paper, but our projects are a special case. You might see this in a paper that developed a simulator for running virtual experiments. If you had no problems, state "No significant problems were encountered." Syntax errors and typo's are not problems. You can give a brief description of how you fixed the problem, for example "Updating to the latest version of java corrected the issues with program faults." Use your best judgement on what is a problem and what is not. If you got help with your problem, acknowledge it here or cite the help as a conversation[1].

V. Experimental Analysis

IF you wish, you can include a paragraph about the specifications of the platform you used to run your experiments. Be very brief: "These timings were carried out on a 427 cubic inch supercharged Dell MT708 with dual overhead BlueRay drives. The power of the CUP was wasted by running Windows 8.1 Pro with all the latest updates as of January 12, 1995. Oracle Java 1.0 was used as the programming language/compiler."

Here you need to graph your results and explain them. This is the main part of your paper. Graphs should be inserted as figures and referenced by "...agreed well with the expected results, see Figure 1. Be careful of color-coding parts of graphs as most printing is still black and white. Notice how I used both color and dashes to differentate the two lines on the graph. You should be able to get away with one graph with two lines. Mark n_0 on the graph if you can.

Current style is to place tables *before* they are referenced with the caption above the table. Figures are place *after* they are referenced and the caption is below the figure. If you have time, read the IEEE Style Sheet in the same folder with this sample.

Explain clearly what your results mean. If you have a result that makes no sense, clearly state why it makes no sense. If you can, give a plausible explanation as to what may have cause the strange result. Do not try to hide a bad result.

VI. Conclusions

Here you answer any questions raised in the previous sections. You should end with a wrap–up and firm statement of what a great accomplishment this is.

[1]Look up how to cite conversations. Also avoid footnotes.

Appendix

An Appendix starts on a new page after everything else. This is were your code will be.

```
  //Sample routine with assert statements
  public void Build_Max_Heap(<T> A[])
   {
//Author:                          G Howser
//Purpose:                 Builds a Max Heap in A[] with root A[1]
//Heap_Test:               Returns "true" iff A[j] Max Heap for j, j+1, j+2, ... n
//
//Pre condition:           A[] is a non-empty array of integers
//Invariant:               A[j] is root of a Max Heap for j=i+1, i+2, ... n
    if (debug)
     {
      assert (((A[1] > 0) || (A[1] <= 0)));
     };
    int heap_size;
    int n = A.length - 1;
    heap_size = A.length - 1; //NOTE:  zero relative arrays but we are ignoring A[0]
    for (int i = Math.floor((n / 2.0)); i--; i < 1)
     {
      Max_Heapify(A, i);
      if (debug)
       {
        assert ((Heap_Test(A[], i));
       };
//Post condition:  A[] is a Max Heap
      if (debug)
       {
        assert ((Heap_Test(A[], 1));
       };
         }
```