Project Submission Details

Please read the following instructions carefully since any significant deviations can result in loss of points.

Part 1

Your submission should consist of a short informal description of what your algorithm does AND a pseudocode for your algorithm with analysis for the running time. You might include an example to show how your algorithm works. You should submit a printed copy for part 1 of the project in class on November 27.

Part 2

Your submission should consist of an encapsulation of files which you should email directly to the TA (eiman@cs.umd.edu). Your program will be graded based on how it runs on the CS cluster or WAM account. It does not matter how it runs on your PC. You should test it to make sure that it runs correctly on CS cluster or WAM account before submission. your encapsulation MUST include the following:

1. makefile: You MUST provide a makefile and the TA should just have to enter "make" to compile your program.
2. README: There must be a file called README, which contains any helpful information to the TA on how to compile and run your program. This includes:
   (a) Your name.
   (b) How to run: Explain how to execute your program on a given input file.
   (c) Known Bugs and Limitations: List any known bugs, deficiencies, or limitations with respect to the project specifications.
3. Source files: (Do not include system files that are available on your platform)

DO NOT INCLUDE: please delete all executable and object files prior to submission.
To submit, store everything (README,makefile, source) in a directory whose name is your first name or something easily identifiable e.g. John. In particular, DO NOT send directories with names like "project". Encapsulate everything into one file. To do that: tar your directory by (tar -cvf John.tar) and then compress the tar file using gzip (gzip John.tar). Send the result as e.mail attachment to the TA (eiman@cs.umd.edu).

Input Format

you will read the input from a file. The input consists of the number of nodes, the source node, the destination node, the cost bound and a description of the graph as follows:

- Number of vertices N
- The cost bound C
- The source s
- The destination t
- The graph itself as quadruples (i, j, l, c). That is an edge from i to j with length l and cost c.

For example:

```
5
120
1
10
(1,2,5,20)
(2,3,7,11)
(1,5,10,10)
...etc
```

Output Format

Your output MUST be in the following format as an output file. Assume that your shortest path is from node s to node i then node $\cdots$ then node k then node t. Let l be the length between the two nodes and c be the cost for this edge. Then your output MUST look like this:

```
(s,i,l,c)
(i,j,l,c)
......
(r,k,l,c)
(k,t,l,c)
```