This assignment continues the thread of *JJ's Jolly Jumping Journey*, or *J5*, from previous ones. The primary **goal** is to use simple graph algorithms as the context for gaining further experience in mapping concrete problems to abstract ones, finding solutions to the abstract problems (using known solutions from books, papers, and other sources), and implementing the solutions in a concrete context. Secondary goals are gaining more experience with programming, documenting algorithms and programs, studying performance, and conducting and summarizing experiments.

The main problem JJ has reached the sharky sea of sinking slides. The sea has numerous small islands, connected by slides. Each slide connects two islands. The slides are barely afloat. Swimming is not an option due to sharks, so the only way to travel is using the slides. By zipping across a slide quickly, JJ can traverse it, but only once because it sinks to the bottom of the sea immediately upon such use. At the middle of each slide is a precious jewel. The *main task* is figuring out the maximum number of jewels that JJ can collect without getting stranded, and returning to the starting position. The input specifies JJ's starting position and the configuration of slides and islands. The desired output is a sequence of islands and slides that JJ can visit in the order provided to return to the initial position and collecting as many jewels as possible. (There should be no solution that allows JJ to collect more jewels.)

Questions

1. (1 pt.) Write your name in the space provided above.
2. (9 pts.) Provide an abstract formulation of the main problem, using familiar mathematical concepts that are independent of JJ's journey or any other specific application. Describe the abstract formulation as precisely and as concisely as possible. Indicate how a solution to the abstract problem may be used to solve JJ's specific problem above.

3. (10 pts.) Describe an efficient algorithm for solving the problem of Question 2. Describe the algorithm in English as precisely as possible. Clearly indicate how the algorithm uses widely known solutions to the problem of Question 2, its subproblems, or related problems. Provide suitable citations for such work.

4. (10 pts.) Explain why the algorithm of Question 3 is correct.

5. (10 pts.) Provide pseudocode, using the textbook's style as a guide, for the algorithm of Question 3. Include explanatory comments and outline a proof of its correctness.

6. (10 pts.) State and justify the running time of the algorithm of Question 5 as a function of the number s of slides (and, optionally, the number b of islands).

7. (150 pts.) Implement the algorithm. Test and document your work carefully and submit your packaged source code and supporting documentation.
8. (20 pts.) Conduct a brief experimental study of your implementation, measuring the running time for a suitable collection of inputs. Include your test code in your electronic submission, with suitable documentation.
9. (30 pts.) Summarize your experimental results by making effective use of charts and tables. Comment on how well the experimental results match the predictions based on your answer to Question 6. Highlight any significant differences and explain them the best you can. Include these results, comments, and explanations as a single PDF file in your submission.

IO format Your program should read from *standard input* and write to *standard output*. The **input** is a sequence of $2m$ whitespace-delimited integers, to be interpreted as m disjoint pairs of adjacent integers. Each pair of numbers denotes a slide between islands identified by the two integers. The very first integer is JJ's starting position. Your program's **output** should consist of one or more lines. The first line consists of just one integer j, which is the maximum number of jewels that can be collected subject to the constraints of the problem. This line is followed by $j + 1$ lines, each of which has a single number that identifies an island. Visiting the islands in the order denoted by this listing of islands should satisfy the constraints of the problem.

Example If the input is

```
1 2 2 3 1 3 2 4 5 4
```

then the desired output is

```
3
1
2
3
1
```

Submission: Follow the submission procedure used for earlier homeworks, putting `hw03` in the obvious place in the file-name.

Reminders Recall, from the previous assignment, policies on collaboration and the use of external resources. Ask for clarifications if anything is unclear. The suggestions in the previous assignment apply to this one too. Use the newsgroup for all questions and discussions unless the matter is private.