A Comprehensive Project for CS2

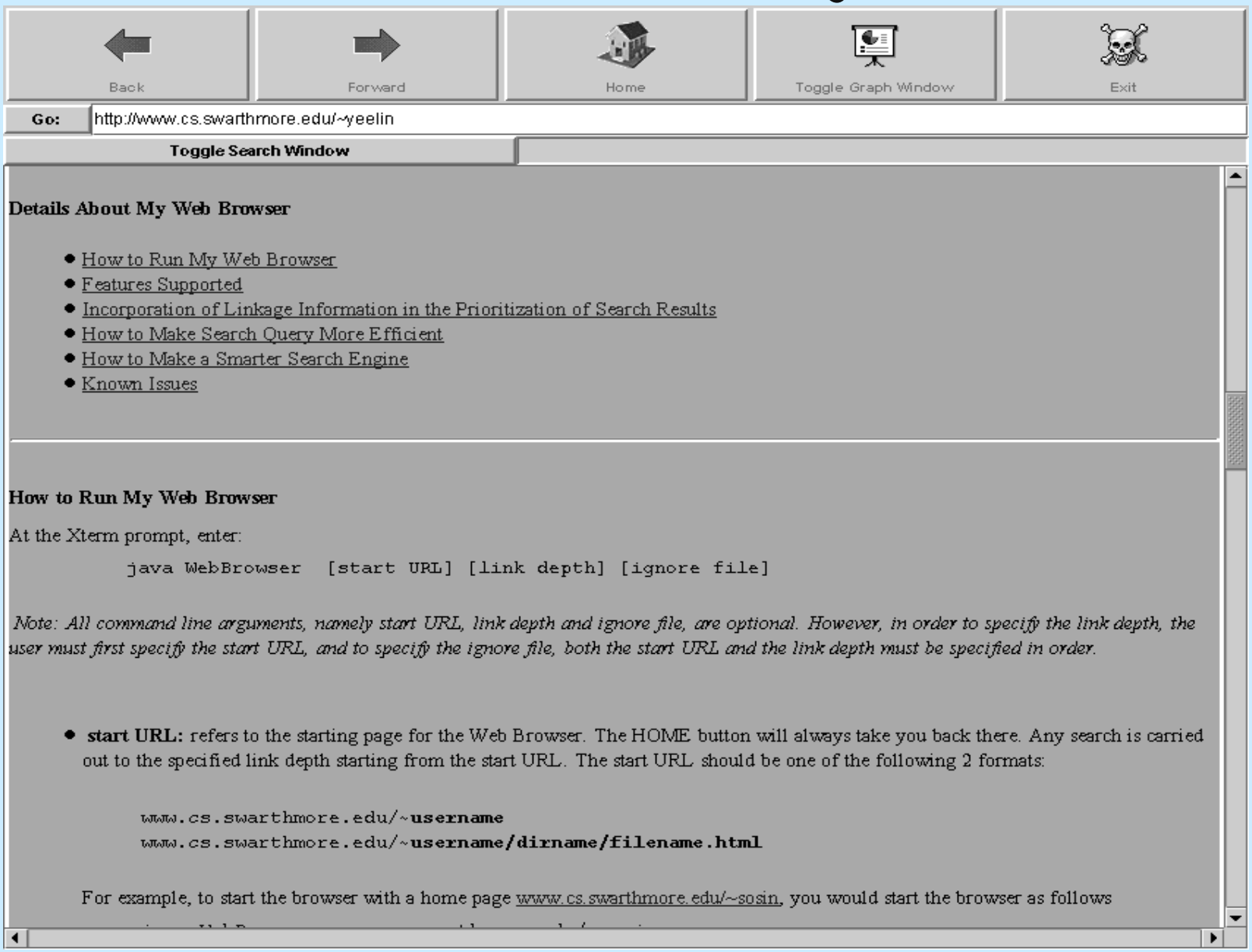
Tia Newhall and Lisa Meeden

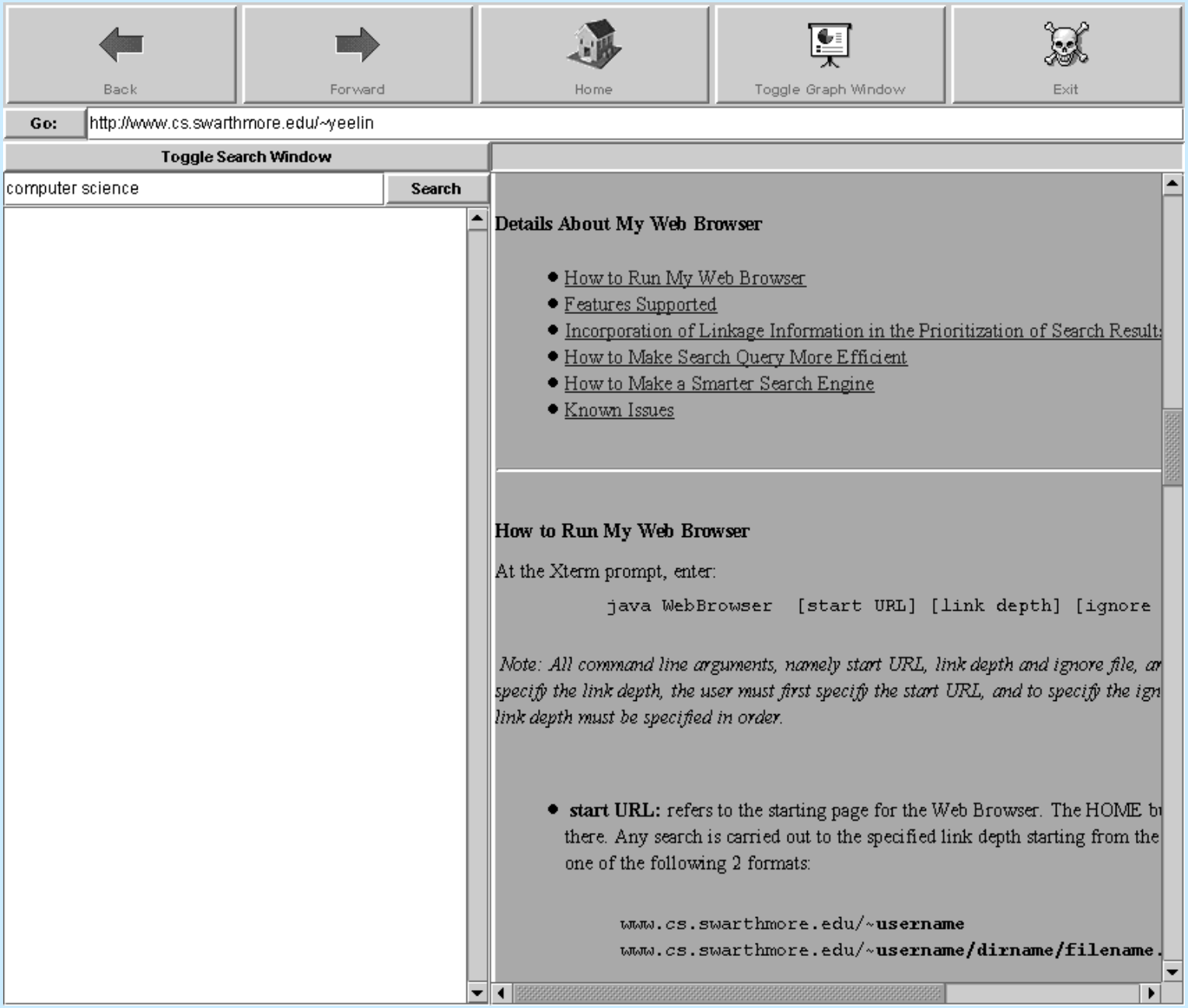
Computer Sciences
Swarthmore college
{newhall, meeden}@cs.swarthmore.edu

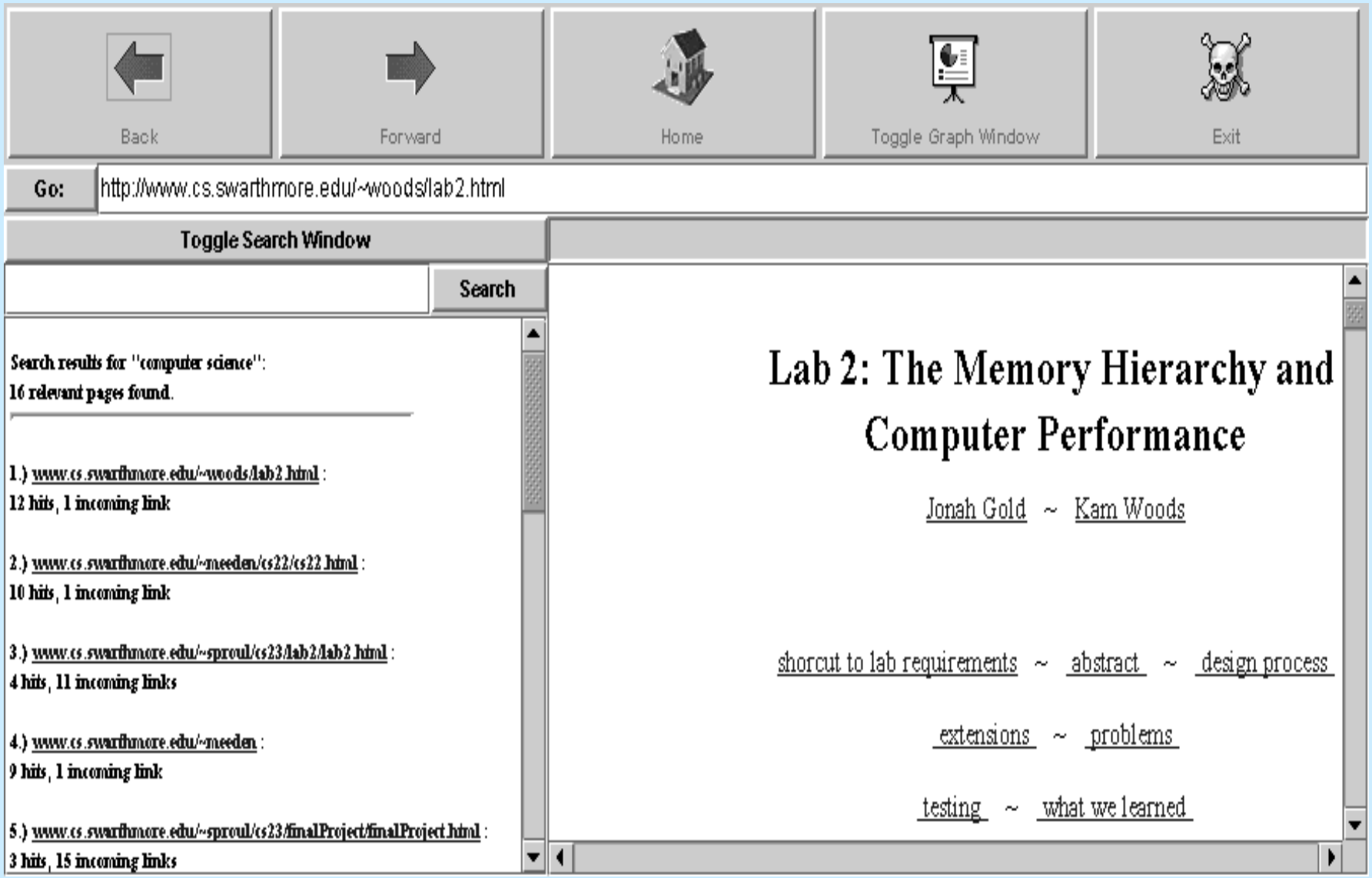
The Project

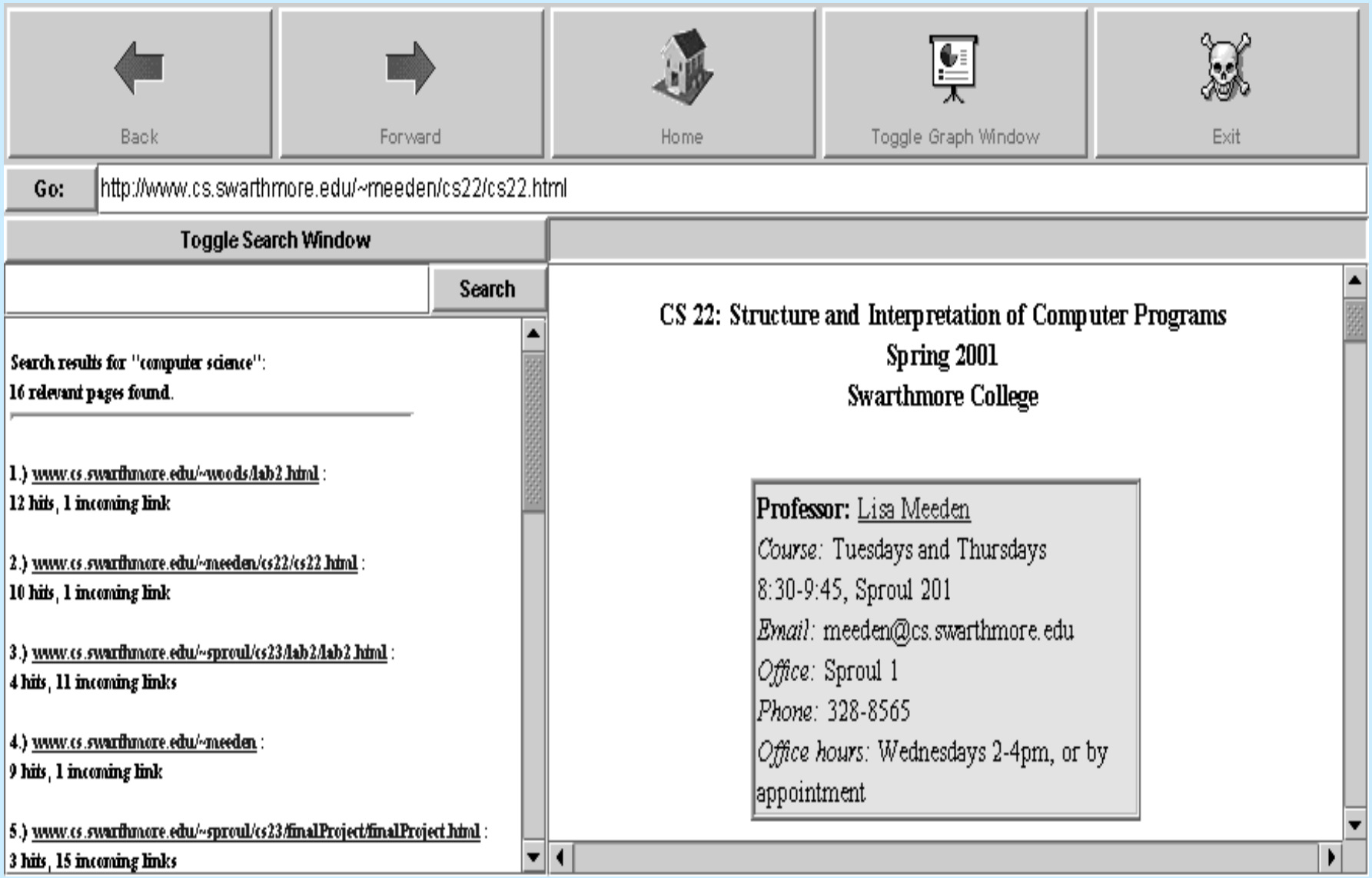
- A Web Browser and Search Engine
  - Find most relevant web pages given a query
  - Automatically display best matching URL
  - Display a web page given a URL
  - Display connectivity information of web pages
  - Answer questions about connectivity of web pages

A Student Project



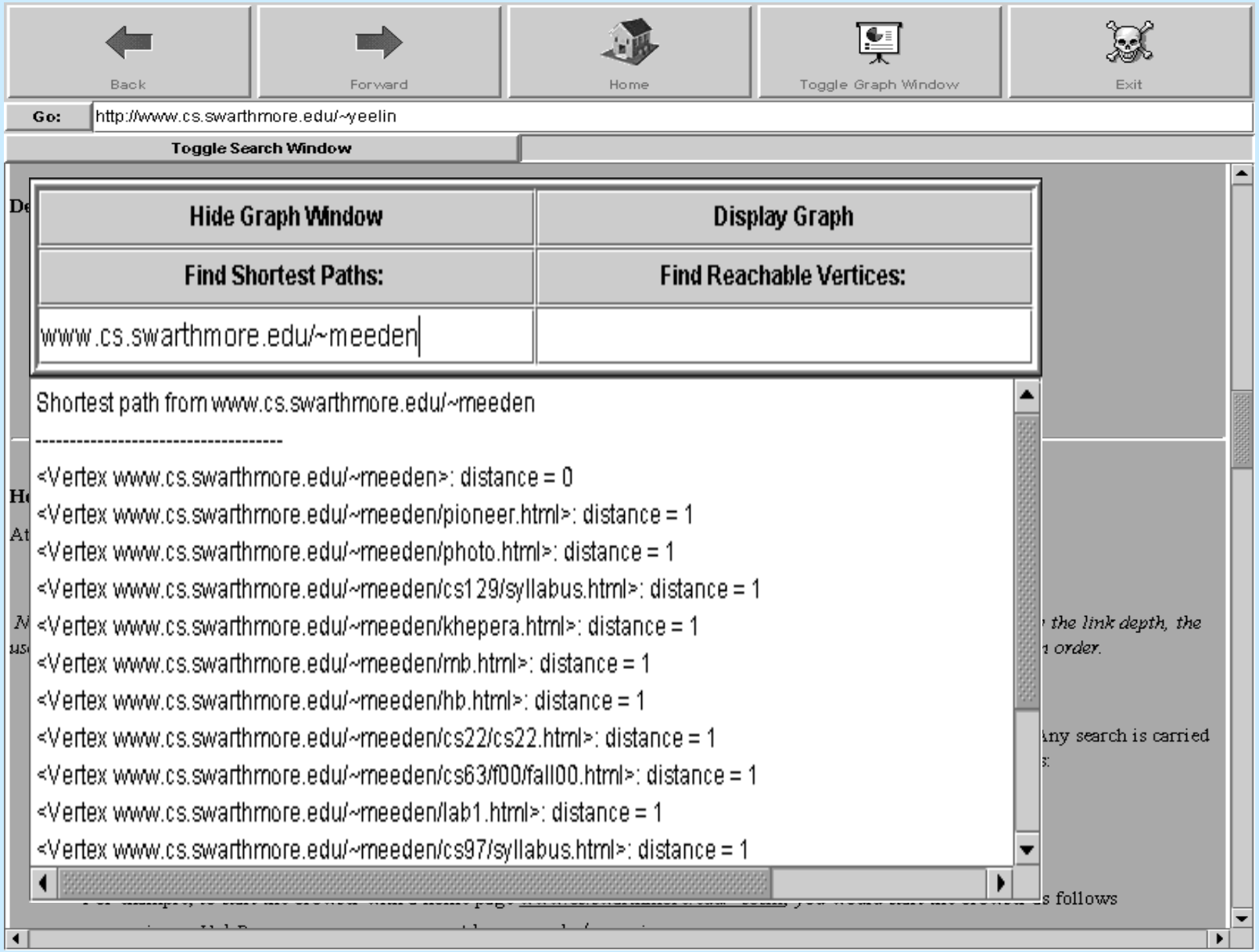

Back
Forward
Home
Toggle Graph Window
Exit
Go:
http://www.cs.swarthmore.edu/~yeelin
Toggle Search Window
computer science
Search
Details About My Web Browser
How to Run My Web Browser
Features Supported
Incorporation of Linkage Information in the Prioritization of Search Result
How to Make Search Query More Efficient
How to Make a Smarter Search Engine
Known Issues
How to Run My Web Browser
At the Xterm prompt, enter:
java WebBrowser [start URL] [link depth] [ignore
Note: All command line arguments, namely start URL, link depth and ignore file, ar
specify the link depth, the user must first specify the start URL, and to specify the ign
link depth must be specified in order.
start URL: refers to the starting page for the Web Browser. The HOME b
there. Any search is carried out to the specified link depth starting from the
one of the following 2 formats:
www.cs.swarthmore.edu/~username
www.cs.swarthmore.edu/~username/dirname/filename.


Back
Forward
Home
Toggle Graph Window
Exit
Go:
http://www.cs.swarthmore.edu/~woods/lab2.html
Toggle Search Window
Search
Search results for "computer science":
16 relevant pages found.
1.) www.cs.swarthmore.edu/~woods/lab2.html :
12 hits, 1 incoming link
2.) www.cs.swarthmore.edu/~meeden/cs22/cs22.html :
10 hits, 1 incoming link
3.) www.cs.swarthmore.edu/~sproul/cs23/lab2/lab2.html :
4 hits, 11 incoming links
4.) www.cs.swarthmore.edu/~meeden :
9 hits, 1 incoming link
5.) www.cs.swarthmore.edu/~sproul/cs23/finalProject/finalProject.html :
3 hits, 15 incoming links
Lab 2: The Memory Hierarchy and Computer Performance
Jonah Gold ~ Kam Woods
shorcut to lab requirements ~ abstract ~ design process
extensions ~ problems
testing ~ what we learned

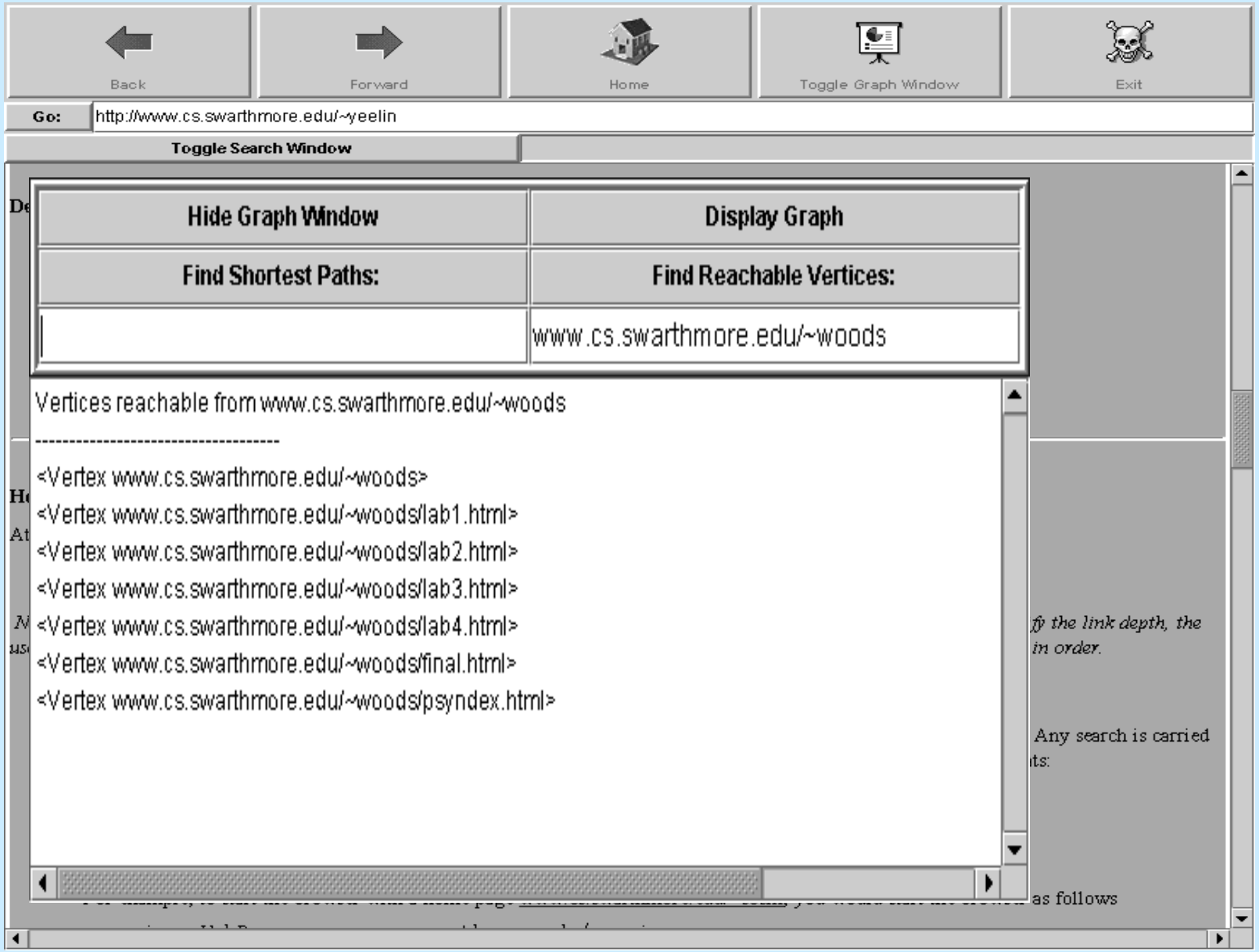

Back
Forward
Home
Toggle Graph Window
Exit
Go:
http://www.cs.swarthmore.edu/~meeden/cs22/cs22.html
Toggle Search Window
Search
Search results for "computer science":
16 relevant pages found.
1.) www.cs.swarthmore.edu/~woods/lab2.html :
12 hits, 1 incoming link
2.) www.cs.swarthmore.edu/~meeden/cs22/cs22.html :
10 hits, 1 incoming link
3.) www.cs.swarthmore.edu/~sproul/cs23/lab2/lab2.html :
4 hits, 11 incoming links
4.) www.cs.swarthmore.edu/~meeden :
9 hits, 1 incoming link
5.) www.cs.swarthmore.edu/~sproul/cs23/finalProject/finalProject.html :
3 hits, 15 incoming links
CS 22: Structure and Interpretation of Computer Programs
Spring 2001
Swarthmore College
Professor: Lisa Meeden
Course: Tuesdays and Thursdays
8:30-9:45, Sproul 201
Email: meeden@cs.swarthmore.edu
Office: Sproul 1
Phone: 328-8565
Office hours: Wednesdays 2-4pm, or by appointment

Back
Forward
Home
Toggle Graph Window
Exit
Go:
http://www.cs.swarthmore.edu/~yeelin
Toggle Search Window
Hide Graph Window
Display Graph
Find Shortest Paths:
Find Reachable Vertices:
Graph

<Graph
<Vertex www.cs.swarthmore.edu/~sproul> visited
<Edge www.cs.swarthmore.edu/~sproul-->www.cs.swarthmore.edu/~sproul/ghettoBrowser/ghetto.html
<Edge www.cs.swarthmore.edu/~sproul-->www.cs.swarthmore.edu/~sproul/cs23/lab2/lab2.html>
<Edge www.cs.swarthmore.edu/~sproul-->www.cs.swarthmore.edu/~sproul/cs23/lab3/lab3.html>
<Edge www.cs.swarthmore.edu/~sproul-->www.cs.swarthmore.edu/~sproul/cs23/finalProject/finalProje
<Vertex www.cs.swarthmore.edu/~sproul/ghettoBrowser/ghetto.html> visited
<Edge www.cs.swarthmore.edu/~sproul/ghettoBrowser/ghetto.html-->www.cs.swarthmore.edu/~sproul
<Edge www.cs.swarthmore.edu/~sproul/ghettoBrowser/ghetto.html-->www.cs.swarthmore.edu/~meed
<Edge www.cs.swarthmore.edu/~sproul/ghettoBrowser/ghetto.html-->www.cs.swarthmore.edu/~newha
<Edge www.cs.swarthmore.edu/~sproul/ghettoBrowser/ghetto.html-->www.cs.swarthmore.edu/~sproul


Back
Forward
Home
Toggle Graph Window
Exit
Go:
http://www.cs.swarthmore.edu/~yeelin
Toggle Search Window
Hide Graph Window
Display Graph
Find Shortest Paths:
Find Reachable Vertices:
www.cs.swarthmore.edu/~meeden
Shortest path from www.cs.swarthmore.edu/~meeden

<Vertex www.cs.swarthmore.edu/~meeden>: distance = 0
<Vertex www.cs.swarthmore.edu/~meeden/pioneer.html>: distance = 1
<Vertex www.cs.swarthmore.edu/~meeden/photo.html>: distance = 1
<Vertex www.cs.swarthmore.edu/~meeden/cs129/syllabus.html>: distance = 1
<Vertex www.cs.swarthmore.edu/~meeden/khepera.html>: distance = 1
<Vertex www.cs.swarthmore.edu/~meeden/mb.html>: distance = 1
<Vertex www.cs.swarthmore.edu/~meeden/hb.html>: distance = 1
<Vertex www.cs.swarthmore.edu/~meeden/cs22/cs22.html>: distance = 1
<Vertex www.cs.swarthmore.edu/~meeden/cs63/f00/fall00.html>: distance = 1
<Vertex www.cs.swarthmore.edu/~meeden/lab1.html>: distance = 1
<Vertex www.cs.swarthmore.edu/~meeden/cs97/syllabus.html>: distance = 1


Back
Forward
Home
Toggle Graph Window
Exit
Go:
http://www.cs.swarthmore.edu/~yeelin
Toggle Search Window
Hide Graph Window
Display Graph
Find Shortest Paths:
Find Reachable Vertices:
www.cs.swarthmore.edu/~woods
Vertices reachable from www.cs.swarthmore.edu/~woods

<Vertex www.cs.swarthmore.edu/~woods>
<Vertex www.cs.swarthmore.edu/~woods/lab1.html>
<Vertex www.cs.swarthmore.edu/~woods/lab2.html>
<Vertex www.cs.swarthmore.edu/~woods/lab3.html>
<Vertex www.cs.swarthmore.edu/~woods/lab4.html>
<Vertex www.cs.swarthmore.edu/~woods/final.html>
<Vertex www.cs.swarthmore.edu/~woods/psyndex.html>
Any search is carried
as follows

Structure of our CS2 Course

Weeks	Topic
1, 2	Intro to Java and OO
3	Complexity Analysis
4	Stacks & Queues
5	GUI Programming, Lists
6, 7	Trees
8	Priority Queues
9, 10	Hashing
11, 12	Graphs
13	Searching & Sorting

Project Goals

- Reinforce data structures and algorithms presented in class
- A fun and challenging problem
- Incorporates complexity analysis
- Real world application
- Team Projects

Project Parts

- Part 1: Analyzing the content of a web page
  - binary search trees
- Part 2: Search Engine
  - priority queues
- Part 3: GUI front-end & Cache of Query Results
  - hash table
- Part 4: Hyper-link graph
  - graph

Part 1: Analyzing the content of web pages

- Use binary search tree
- Parse web page & create word frequency tree
- Develop a list of words to be ignored

```
% Java WordFreqTree meeden.html 3 ignore_list
AAAI:  5
AI:  7
AIC:  3
ALAN:  3
ARTIFICIAL:  8
AUTHORED:  13
```

Part 2: Query Processing

- Adds Priority Queues
- Use word frequency trees to calculate a relevance score for each URL
  - Relevance is sum of word frequency counts of each word in query
- URL results listed in order of best match

```
% java url_list ignore_list

Enter Next Query or -1 to quit:  robot
Query Results for "robot"
------------------------------
www.cs.swarthmore.edu/~meeden                    priority: 19
www.cs.swarthmore.edu/~meeden/robots/game.html   priority: 9

Enter Next Query or -1 to quit:  brain
Query Results for "brain"
------------------------------
www.cs.swarthmore.edu/~meeden/robots/game.html    priority: 12
www.cs.swarthmore.edu/~meeden                     priority: 1

Enter Next Query or -1 to quit:  robot brain
Query Results for "robot brain"
------------------------------
www.cs.swarthmore.edu/~meeden/robots/game.html    priority: 21
www.cs.swarthmore.edu/~meeden                     priority: 20
```

Part 3: Cache of query results and GUI front-end

- Adds hash table
- Process a Query
  - Look in cache for complete result
  - Look in cache for partial result and merge together to get full result
- GUI front-end
  - Display web pages
  - Enter query and display URL list of results
  - Automatically display best matching web page

Part 4: Hyperlink Graph

- Adds Graphs
- Build connectivity graph by parsing links to a limited depth on Web pages
- Incorporate linked-to information in ordering query results
- Display information about connectivity graph
  - Shortest path
  - Reachability

Conclusions

- Students: overwhelmingly positive response
  - Fun and Challenging
  - Liked real world project
  - Liked building large application in parts
  - Reinforced data structures and algorithms
- Professors: good vehicle for teaching data structures
  - Incorporates all data structures
  - Applies complexity analysis
- **www.cs.swarthmore.edu/~newhall/sigcse02**