

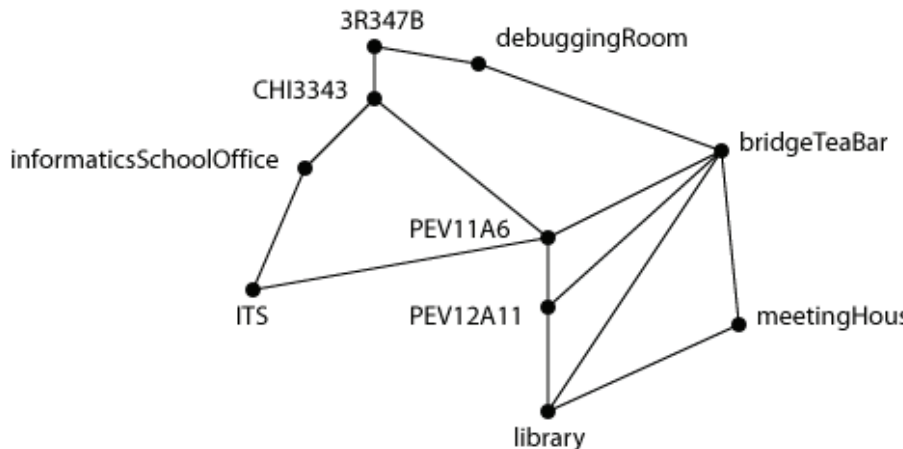
KR-IST - Lecture 2b: Route finding in Java

Chris Thornton

October 30, 2014

Introduction

This lecture looks at a search program for finding routes in a toy map of campus.



CampusSearch class

```
import java.util.*;

public class CampusSearch {

    String links[][] = {
        {"office", "debuggingRoom"},
        {"office", "CHI343"},
        {"CHI343", "informaticsSchoolOffice"},
        {"informaticsSchoolOffice", "ITS"},
        {"ITS", "PEV11A6"},
        {"PEV11A6", "PEV12A11"},
        {"PEV11A6", "library"},
        {"CHI343", "PEV11A6"},
        {"PEV12A11", "library"},
        {"library", "bridgeTeaBar"},
        {"library", "meetingHouse"},
        {"meetingHouse", "bridgeTeaBar"},
        {"bridgeTeaBar", "debuggingRoom"}};

    String goal = "meetingHouse";
```

Successor function

```
ArrayList<String> getSuccessors(String location) {
    ArrayList<String> successors = new ArrayList<String>();
    for (int i = 0; i < links.length; i++) {
        if (links[i][0].equals(location)) {
            successors.add(links[i][1]); }
        if (links[i][1].equals(location)) {
            successors.add(links[i][0]); }
    }
    return(successors);
}
```

Main loop

```
void run() {
    ArrayList<ArrayList<String>> open = new ArrayList<Arra
    ArrayList<String> path = new ArrayList<String>();
    path.add("office");
    open.add(path);

    while (open.size() > 0) {
        path = open.remove(0);
        String parent = path.get(path.size()-1);
        if (parent.equals(goal)) {
            System.out.println("ROUTE: " + path); }
        ArrayList<String> successors = getSuccessors(parent);
        for (int i = 0; i < successors.size(); i++) {
            String child = successors.get(i);
            if (!path.contains(child)) {
                ArrayList<String> newPath = new ArrayList<String>
                newPath.add(child);
                open.add(newPath); }
            }
        }
    }
```

main method

```
public static void main(String args[]) { // do the search
    new CampusSearch().run();
}
}
```

Running the program the traditional way

Compile the program with "javac", then run it with "java" command:

```
javac CampusSearch.java  
java CampusSearch
```

Running it from BlueJ

If you are running Java via BlueJ, then you would run the program by explicitly calling the 'run' method of the CampusSearch class.

You could also explicitly call the 'main' method; i.e., right-click on the CampusSearch box, choose 'void main(args)'. Then press 'Ok'.

Output generated

The program generates all possible routes connecting 'office' with 'meetingHouse', ordered by length.

```
ROUTE: [office, debuggingRoom, bridgeTeaBar, meetingHouse]
ROUTE: [office, debuggingRoom, bridgeTeaBar, library, meetingHouse]
ROUTE: [office, CHI343, PEV11A6, library, meetingHouse]
ROUTE: [office, CHI343, PEV11A6, PEV12A11, library, meetingHouse]
ROUTE: [office, CHI343, PEV11A6, library, bridgeTeaBar, meetingHouse]
ROUTE: [office, CHI343, informaticsSchoolOffice, ITS, PEV11A6, library, meetingHouse]
ROUTE: [office, CHI343, PEV11A6, PEV12A11, library, bridgeTeaBar, meetingHouse]
ROUTE: [office, CHI343, informaticsSchoolOffice, ITS, PEV11A6, PEV12A11, library, meetingHouse]
ROUTE: [office, CHI343, informaticsSchoolOffice, ITS, PEV11A6, library, meetingHouse]
ROUTE: [office, CHI343, informaticsSchoolOffice, ITS, PEV11A6, PEV12A11, library, meetingHouse]
```

Summary

- ▶ CampusSearch class

Summary

- ▶ CampusSearch class
- ▶ Successor function

Summary

- ▶ CampusSearch class
- ▶ Successor function
- ▶ Main loop

Summary

- ▶ CampusSearch class
- ▶ Successor function
- ▶ Main loop
- ▶ main method

Summary

- ▶ CampusSearch class
- ▶ Successor function
- ▶ Main loop
- ▶ main method
- ▶ Running the program the traditional way

Summary

- ▶ CampusSearch class
- ▶ Successor function
- ▶ Main loop
- ▶ main method
- ▶ Running the program the traditional way
- ▶ Running it from BlueJ

Summary

- ▶ CampusSearch class
- ▶ Successor function
- ▶ Main loop
- ▶ main method
- ▶ Running the program the traditional way
- ▶ Running it from BlueJ

Questions

Questions

- ▶ Does the ordering of link data make any difference? Does it make any difference if {"library", "bridgeTeaBar"} comes before {"library", "meetingHouse"} in the array?

Questions

- ▶ Does the ordering of link data make any difference? Does it make any difference if {"library", "bridgeTeaBar"} comes before {"library", "meetingHouse"} in the array?
- ▶ Does it make any difference if {"library", "meetingHouse"} appears as {"meetingHouse", "library"}?

Questions

- ▶ Does the ordering of link data make any difference? Does it make any difference if {"library", "bridgeTeaBar"} comes before {"library", "meetingHouse"} in the array?
- ▶ Does it make any difference if {"library", "meetingHouse"} appears as {"meetingHouse", "library"}?
- ▶ How is the goal for a particular search presented? How could we improve on this?

Questions

- ▶ Does the ordering of link data make any difference? Does it make any difference if {"library", "bridgeTeaBar"} comes before {"library", "meetingHouse"} in the array?
- ▶ Does it make any difference if {"library", "meetingHouse"} appears as {"meetingHouse", "library"}?
- ▶ How is the goal for a particular search presented? How could we improve on this?
- ▶ What advantages are there of using ArrayList data in preference to user-defined data?

Questions

- ▶ Does the ordering of link data make any difference? Does it make any difference if {"library", "bridgeTeaBar"} comes before {"library", "meetingHouse"} in the array?
- ▶ Does it make any difference if {"library", "meetingHouse"} appears as {"meetingHouse", "library"}?
- ▶ How is the goal for a particular search presented? How could we improve on this?
- ▶ What advantages are there of using ArrayList data in preference to user-defined data?
- ▶ What does the ArrayList "remove" method do? Suggest a more meaningful name for this method.

Questions

- ▶ Does the ordering of link data make any difference? Does it make any difference if {"library", "bridgeTeaBar"} comes before {"library", "meetingHouse"} in the array?
- ▶ Does it make any difference if {"library", "meetingHouse"} appears as {"meetingHouse", "library"}?
- ▶ How is the goal for a particular search presented? How could we improve on this?
- ▶ What advantages are there of using ArrayList data in preference to user-defined data?
- ▶ What does the ArrayList "remove" method do? Suggest a more meaningful name for this method.
- ▶ What strategy does this program use for purposes of producing solution paths?

Questions

- ▶ Does the ordering of link data make any difference? Does it make any difference if {"library", "bridgeTeaBar"} comes before {"library", "meetingHouse"} in the array?
- ▶ Does it make any difference if {"library", "meetingHouse"} appears as {"meetingHouse", "library"}?
- ▶ How is the goal for a particular search presented? How could we improve on this?
- ▶ What advantages are there of using ArrayList data in preference to user-defined data?
- ▶ What does the ArrayList "remove" method do? Suggest a more meaningful name for this method.
- ▶ What strategy does this program use for purposes of producing solution paths?

Adapt the CampusSearch program for use with data derived from the toy map of Brighton featured in the previous exercises. Check that the program produces valid routes for that map.

Adapt the CampusSearch program for use with data derived from the toy map of Brighton featured in the previous exercises. Check that the program produces valid routes for that map.

- ▶ Estimate the space complexity for this program searching a tree of depth 4 and branching factor 3.

Adapt the CampusSearch program for use with data derived from the toy map of Brighton featured in the previous exercises. Check that the program produces valid routes for that map.

- ▶ Estimate the space complexity for this program searching a tree of depth 4 and branching factor 3.
- ▶ Modify the program so that it produces only the shortest route between two locations.

Adapt the CampusSearch program for use with data derived from the toy map of Brighton featured in the previous exercises. Check that the program produces valid routes for that map.

- ▶ Estimate the space complexity for this program searching a tree of depth 4 and branching factor 3.
- ▶ Modify the program so that it produces only the shortest route between two locations.
- ▶ Modify the program so that it prints out the total number of locations inspected in order to obtain the shortest route between two locations.

Adapt the CampusSearch program for use with data derived from the toy map of Brighton featured in the previous exercises. Check that the program produces valid routes for that map.

- ▶ Estimate the space complexity for this program searching a tree of depth 4 and branching factor 3.
- ▶ Modify the program so that it produces only the shortest route between two locations.
- ▶ Modify the program so that it prints out the total number of locations inspected in order to obtain the shortest route between two locations.

Exercises cont.

- ▶ Modify the program so that it performs depth-first search if the value of boolean class variable "dfs" is `<true>`, and breadth-first search otherwise.

- ▶ Modify the program so that it performs depth-first search if the value of boolean class variable "dfs" is `<true>`, and breadth-first search otherwise.
- ▶ Modify the program so that it searches to a maximum depth of four levels in the search tree.

Exercises cont.

- ▶ Modify the program so that it performs depth-first search if the value of boolean class variable "dfs" is `<true>`, and breadth-first search otherwise.
- ▶ Modify the program so that it searches to a maximum depth of four levels in the search tree.
- ▶ Modify the program so that the depth-limit may be passed in as a command-line argument. (This will involve modifying the "main" method.)

Exercises cont.

- ▶ Modify the program so that it performs depth-first search if the value of boolean class variable "dfs" is `<true>`, and breadth-first search otherwise.
- ▶ Modify the program so that it searches to a maximum depth of four levels in the search tree.
- ▶ Modify the program so that the depth-limit may be passed in as a command-line argument. (This will involve modifying the "main" method.)