BLG336E - Analysis of Algorithms II 28.02.2018

2017–2018 Spring, Project 1

Submission Deadline: 18.03.2018 23:00

Overview

Breadth First Search (BFS) and Depth First Search (DFS) are well-known graph traverse algorithms. In this project, you will need to implement BFS and DFS algorithms in order to solve the given problem.

Farmer, Rabbit, Fox and Bag of Carrots

A farmer with his rabbit, bag of carrots and fox came to the east side of a river that they wish to cross. The farmer has a boat at the river's edge, but only he can row. The boat can only hold single thing(rabbit, bag of carrots or fox) in addition to the rower at any one time. If the fox is left alone with the rabbit, the fox may eat it. Similarly if the rabbit is left alone with the bag of carrots, the rabbit will eat it. How can the farmer get to the west side of the river so that all four arrive safely?

A) Implementation

1) Formalize this problem in a well-defined graph form and present your state and action representations in detail.

2) Run both BFS and DFS algorithms and analyze the results in terms of:
   - the number of the visited nodes
   - the maximum number of nodes kept in the memory
   - the running time.
   - the number of move steps on the found solution.

3) The algorithm (BFS or DFS) to be used for search should be chosen by a command line argument as in the given example.

4) When the solution is found, you should print the sequence of moves so that all four arrive safely to the other side of the river.

Your program should compile and run using the following commands:

```
g++ sourcecode.cpp –o project1
./project1 bfs
./project1 dfs
```

Sample Output: The numbers may not reflect the real results.

```
> ./project1 dfs
Algorithm: DFS
Number of the visited nodes: 21
Maximum number of nodes kept in the memory: 9
Running time: 0.34 seconds
Solution move count: 9
Farmer Rabbit Carrot Fox ||
(Farmer, Rabbit, > )
Carrot Fox || Farmer Rabbit
(Farmer, <)
Farmer Carrot Fox || Rabbit
(Farmer, Carrot, > )
Fox || Farmer Carrot Rabbit
...
|| Farmer Rabbit Carrot Fox
```

B) Report

1) Present your problem formulation, state and action representations in detail.

2) How does your algorithms work?
    a) Write your pseudo-code.
    b) Show complexity of your algorithm on pseudo-code.

3) In Depth-First Search algorithm, why should we maintain a list of discovered nodes?

4) Is the graph constructed by the given problem formulation a bipartite graph? Why or why not? (Implementation is **not** required for this part.)

5) Analyze and explain the algorithm results in terms of:
    - the number of visited nodes
    - the maximum number of nodes kept in the memory
    - the running time.
    - the number of move steps on the found solution.

Note: If you have any questions, please feel free to contact Res. Asst. Çağatay Koç via e-mail (kocca@itu.edu.tr).

Policy:

- You may discuss the problem addressed by the project at an abstract level with your classmates, but you should not share or copy code from your classmates or from the Internet.You should submit your own, individual project.
- **Academic dishonesty including but not limited to cheating, plagiarism, collaboration is unacceptable and subject to disciplinary actions.**

Submission Instructions:

- Please submit your homework through Ninova e-Learning System.
- You must submit **all your source code in a single cpp file** and **a softcopy report (PDF).** You can define multiple classes in a single cpp file.
- All your code must be written in C++, and we must be able to compile and run on it on ITU's Linux Server (you can access it through SSH) using g++.
- When you write your code, try to follow an object-oriented methodology with well-chosen variable, method, and class names and comments where necessary.
- Your code must compile without any errors; otherwise, you may get a grade of zero on the assignment.
- You should be aware that the Ninova e-Learning System clock may not be synchronized with your computer, watch, or cell phone. Do not e-mail the teaching assistant or the instructors your submission after the Ninova site submission has closed. If you have submitted to Ninova once and want to make any changes to your report, you should do it before the Ninova submission system closes. Your changes will not be accepted by e-mail. Connectivity problems to the Internet or to Ninova in the last few minutes are not valid excuses for being unable to submit. You should not risk leaving your submission to the last few minutes. After uploading to Ninova, check to make sure that your project appears there.