Daniel Campos

October 18th, 2012

Algorithms Section 5

HW 4

1. Problem 3.7

a) An undirected graph is bipartite if it can be colored by two colors. The algorithm we present is a modiﬁed DFS that colors the graph using 2 colors.

function graph-coloring(G)

Input: Graph G

Output: returns true if the graph is bipartite false otherwise

for all v ∈ V:

visited(v)= false

color(v) = GREY

while ∃s ∈ V : visited(s) = false

visited(s) = true

color(s) = WHITE

S = [s] (stack containing v)

while S is not empty

u = pop(S)

for all edges (u,v) ∈ E:

if visited(v) = false:

visited[v] = true

push(S,v)

if color(v) = GREY

if color(u) = BLACK:

color(v) = WHITE

if color(u) = WHITE:

color(v) = BLACK

else if color(v) = WHITE:

if color(u) =! BLACK:

return false

else if color(v) = BLACK:

if color(u) =! WHITE:

return false

return true

B) Consider a path P whose start vertex is s, end vertex is t and it passes through vertices u1, u2, ..., un and the associated edges are (r, u1),(u1, u2), ...,(un, p).

If P is a cycle, then r and p are the same vertices. assume s is in V1. Each edge (ui, ui+1) goes from one vertex set to other. The path must have 2·i edges to come back into the same vertex set where i ∈ N. Since r and p are in same vertex set, so the length of the cycleformed must be 2·i which is even.

Odd length: the cycle C passes through vertices u1, u2, ..., un where u1 = un. The associated edges are (u1, u2), ...,(un−1, un).

We color edges with two colors WHITE and BLACK. Without any loss of generality u1 is colored WHITE while u^n−1 is colored BLACK since n is odd and therefore n − 1 is even. Choosing color of u^n as WHITE conﬂicts with the color of u^n−1 while choosing color as BLACK conﬂicts with the color of u1. Therefore it is not possible to color an odd cycle with 2 colors which implies that the graph is not bipartite

C)3

1. Problem 3.8

A) A state transition graph, in which each node indicates a possible

distribution of water in the three containers. Each state is represented by a tuple (a, b, c) that indicates the 10-pint container has a pint water, the 7-pint container has b point water and the 4-pint container has c pint water. We create an edge from vertex (a, b, c) to vertex (x, y, z) if by pouring water one container into another implements the state transition.

B) The algorithm to be used would be a depth first search hoping to find (0, 7, 4) to (x, 2, z) and (x, y, 2).

C) Extra\*\*\*\*\*\*\*\*\*

A path may be

(0, 7, 4) ->(4, 7, 0)->(10, 1, 0) ->(6, 1, 4) ->(6, 5, 0) ->(2, 5, 4) ->(2, 7, 2)

1. Problem 3.11

Function contains\_e

Input: Undirected Graph G and edge E={e,v} in it

Run explore(G – E,u)//Explore is shown bellow

If node v gets visited

Return true

Else

Return false

function explore(G, v)

create an empty statck S

for all u in V

visited(u) = false

visited(v) = true

previsit(v)

push(S, v)

while S is not empty

p = pop(S)

if all the neighbors of p are visited

postvisit(p)

else

push(S, p)

for each edge (p,q) in E

if not visited(q)

visited(q) = true

previsit(q)

push(S, q)

Reasoning

G has a cycle containing e if and only if graph G without edge e has a path from u to v. Removing edge takes linear time. The rest of the algorithm runs in linear time because DFS does.

1. Problem 4.1

A)

B)

A 1 B 2 C 1 D

4 4

E F 1 G 1 H

1. Problem 4.8

No By counter example

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| A | B | C | D | E | F | G | H |
| **0** |  |  |  |  |  |  |  |
| 0 | **1** |  |  | 4 | 8 |  |  |
| 0 | 1 | **3** |  | 4 | 7 | 7 |  |
| 0 | 1 | 3 | **4** | 4 | 7 | 5 |  |
| 0 | 1 | 3 | 4 | **4** | 7 | 5 | 8 |
| 0 | 1 | 3 | 4 | 4 | 7 | **5** | 8 |
| 0 | 1 | 3 | 4 | 4 | **6** | 5 | 6 |
| 0 | 1 | 3 | 4 | 4 | 6 | 5 | **6** |
| 0 | 1 | 3 | 4 | 4 | 6 | 5 | 6 |

Consider a graph where there are two paths from point A to B. One going a single length of 3 and the other path going through the edges with values 1,2,-2. The second path would take 1 where the first would take 3 by adding a 4 to make all of them positive the second path would now be length 13 vs. the first path being 7. The tactic of increasing the values only works if the amount of steps needed is equal.