Discrete Mathematics   Quiz 2

2006.05.09

Name:______________________________ Student ID:______________________________

Instructions. This is a 2-hr close book quiz. Please manage your time well. No dictionary, calculator, PDA, or any other electronic device. Any dishonorable cheating behavior will give you a miserable future. (Totally 115 points)

1. [40%] The Floyd-Warshall Algorithm is described in the last page. Given a digraph as in Fig 1

   (a) [14%] Apply Floyd-Warshall Algorithm to compute the optimal distance matrix D, and predecessor matrix PRED

C=

0	3	-2	∞	∞
∞	0	4	1	∞
∞	∞	0	-2	-3
∞	∞	∞	0	-2
6	∞	∞	∞	0

PRED=

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

D^1

D^2

D^3

D^4

D^5

$PRED^1$

$PRED^2$

$PRED^3$

$PRED^4$

$PRED^5$

   (b) [8%] Trace the shortest paths from 1 to 5, 2 to 1, 4 to 5, and 5 to 4

   (c) [3%] Suppose D^4_{34} is the $(3,4)^{th}$ entry of the Distance matrix after the 4^{th} iteration of F-W algorithm, explain what D^4_{34} means

   (d) [3%] List all paths of 4 edges from 4 to 2

   (e) [3%] List all paths of 3 edges from 1 to 5

   (f) [6%] Compute $C^2_{2,3}$ and $C^2_{4,3}$

   (g) [3%] Explain what $C^k_{i,j}$ means

   Ans:

2. [30%] Given an undirected graph G_1 as shown in Fig. 2, answer the following questions:
   (a) [10%] Starting from vertex 1, draw a Breadth-First-Search tree
   (b) [10%] Using Prim's algorithm, draw a Minimum Spanning Tree of G_1
   (c) [10%] Starting from vertex 1, using Dijkstra's algorithm to draw a 1-ALL shortest path tree

   Ans:

3. [10%] Suppose you have used Dijkstra's Algorithm to solve a 1-ALL shortest path problem for a directed graph $G = (V, E)$ and computed the optimal distance label $d[i]$, $\forall i \in V$. (Answers without explanation get at most 1 point)
(a) [6%] Suppose each edge (i, j) has a length c_{ij}, give a method to check whether your answers $d[i]$ are correct $\forall i \in V$.
Note that you are **NOT** allowed to use another shortest path algorithm to solve the same problem for verifying whether your original solution is correct or not.
(b) [4%] What is the complexity of your method in (a)?

Ans:

4. [10%] Given a connected graph $G = (V, E)$, answer the following questions (Answers without explanation get at most 1 point)
(a) [6%] Give a method to check whether G contains an odd cycle (hint: using the BFS algorithm, then....?)
(b) [4%] What is the complexity of your method?

Ans:

5. [10%] For a graph $G = (V, E)$ with $|V| = n$ vertices, $|E| = m$ edges, and $C = \max\{c_{ij} : (i,j) \in E\}$, give the complexity (e.g. $O(n^2)$, $O(m \log n)$,...,etc.) of the individual procedures (**vertex selection** and **distance update)** and the overall complexity for the following implementations of Dijkstra's algorithm. Explain your answers.
(a) [3%] Naive implementation (using linear search to select the vertex with minimum distance label)
(b) [4%] Binary min-heap implementation
(c) [3%] Dial's bucket implementation

Ans:

6. [10%] True or False, and EXPLAIN
Circle T or F for each of the following statements to indicate whether the statement is true or false, respectively. If the statement is correct, briefly state why. If the statement is wrong, explain why or give a counter example. Answers without reasons will get at most 1 point.
(a) [3%] **(T , F)** There are countries on a planar map, adjacent countries must be colored by different colors. Any such a map can be colored by 5 color.
Ans:

(b) [3%] **(T , F)** For a complete bipartite graph $K_{m,n}$ where $10 \leq m < n \leq 15$, since every vertex of this graph has degree at most 15, it is 16-colorable. Thus its chromatic number is 16.
Ans:

(c) [4%] Suppose G contains directed cycles, all arcs have different lengths, some arcs have negative lengths, but there exist no directed cycles with negative lengths. Thus Dijkstra's algorithm can **NOT** be directly applied. Suppose $C = \min_{a \in A}\{c_a\} < 0$. To solve a single source shortest path problem in G from a source node s, someone suggests the following method:

Algorithm Modified_Dijkstra
begin
  for $i = 1$ to m **do**
    $c_i = c_i + |C|$;
  do Dijkstra's algorithm starting from source node s with the updated arc length c
end

Will this method solves the original problem? If yes, explain why; otherwise, explain and give a counter example.
Ans: