Homework #4

Anthony Menjivar

4.2 (a) Bellman-Ford Algorithm

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| # | S | A | B | C | D | E | F | G | H | I |
| 0 | 0 | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ |
| 1 | 0 | 7 | ∞ | 6 | ∞ | 6 | 5 | ∞ | ∞ | ∞ |
| 2 | 0 | 7 | 11 | 5 | 8 | 6 | 4 | ∞ | 9 | ∞ |
| 3 | 0 | 7 | 11 | 5 | 7 | 6 | 4 | 9 | 7 | ∞ |
| 4 | 0 | 7 | 11 | 5 | 7 | 6 | 4 | 8 | 7 | 8 |
| 5 | 0 | 7 | 11 | 5 | 7 | 6 | 4 | 8 | 7 | 7 |

4.8 Counter Example Proof

In the graph below we have u 🡪 v 🡪 w being the shortest from u to w. If we or 10 or more to each edge, the shortest path changes u 🡪 w.

20

-10

**w**

**v**

**u**

15

4.17 (a)

All dist values will be in the range {0, 1, …, W(|V| - I), ∞} because any shortest path contains at most |v| - 1 edges. We can implement heap on dist values by maintaining an array indexed by all possible values of dist where each I is a pointer to a linked list of elements dist value equal to i. With this we can perform insert operations in constant time by appending value at the beginning. Makeheap will take O(|V|) because performing Djikstra’s algorithm the min value increases and deltemin will just require to search for the new min value. Deletemin will take O(W|V|) time since we will look at each array entry at most once.. The deacreasekey operation can be implemented by inserting a new copy of an element into the list corresponding to its new value without removing copies. There will be at most |E| decresekey operations so at most |E| copies which we need to ignore, paying attention only to time O(|E|). Each decresekey take O(1) meanining decresekey takes O(|E|). Therefore, the implementation time takes O(|V|) + O(W|V|) + O(|E|) = O(W|V| + |E|)

5.2 Minimum Spanning Tree

(a) Prim’s Algorithm

|  |  |  |
| --- | --- | --- |
| Vertex | Edge | Cost |
| A |  | 0 |
| B | AB | 1 |
| C | BC | 3 |
| G | CG | 5 |
| D | GD | 6 |
| F | GF | 7 |
| H | GH | 8 |
| E | AE | 12 |

(b) Kruskal’s Algorthm

A

C

E

B

F

H

G

D

D

Kirkman School Girl Solver is in my github repository:

<https://github.com/TonyMenji/CMSI282/tree/master/homework4/Kirkman%20Schoolgirl%20Solver>

SubsetSum.java

