Problem 2:

Problem 24-4 in textbook.

***24-4 Gabow’s scaling algorithm for single-source shortest paths***

A ***scaling*** algorithm solves a problem by initially considering only the highestorder

bit of each relevant input value (such as an edge weight). It then refines the

initial solution by looking at the two highest-order bits. It progressively looks at

more and more high-order bits, refining the solution each time, until it has examined

all bits and computed the correct solution.

In this problem, we examine an algorithm for computing the shortest paths from

a single source by scaling edge weights. We are given a directed graph G=(V,E)

with nonnegative integer edge weights *w*. Let W =. Our

goal is to develop an algorithm that runs in O.E lgW / time. We assume that all

vertices are reachable from the source.

The algorithm uncovers the bits in the binary representation of the edge weights

one at a time, from the most significant bit to the least significant bit. Specifically,

let k D dlg.W C 1/e be the number of bits in the binary representation of W ,

and for i D 1;2; : : : ;k, let wi .u; \_/ D

w.u; \_/=2k\_i

˘

. That is, wi .u; \_/ is the

“scaled-down” version of w.u; \_/ given by the i most significant bits of w.u; \_/.

(Thus, wk.u; \_/ D w.u; \_/ for all .u; \_/ 2 E.) For example, if k D 5 and

w.u; \_/ D 25, which has the binary representation h11001i, then w3.u; \_/ D

h110i D 6. As another example with k D 5, if w.u; \_/ D h00100i D 4, then

w3.u; \_/ D h001i D 1. Let us define ıi .u; \_/ as the shortest-path weight from

vertex u to vertex \_ using weight function wi . Thus, ık.u; \_/ D ı.u; \_/ for all

u; \_ 2 V . For a given source vertex s, the scaling algorithm first computes the