Algorithm 1 delta-stepping GPU SSSP

Input: $G(V, E)$, source vertex s;
Output: $dist(v)$, $(v \in V)$, the weight of the shortest path from s to v;

```
1:
2: function initial(s, V)
3:     for each v ∈ V do
4:         dist(v) ← +∞;                          ▷ initialize dist to positive infinity;
5:     end for
6:     dist(s) ← 0;                                ▷ set the source distence to 0;
7: end function
8:
9: function delta_steppingCudaFunc(G(V,E), dist, predist)   ▷ G(V,E), the initially distance array
   dist, a temporary distance array predist, the delta Δ;
10:    u0 ← threadId;                                ▷ get the thread id;
11:    offset ← blockDim;                            ▷ get the number of threads in a block;
12:    id ← (__shared__ memory) 1;                   ▷ the bucket id;
13:    B ← (__shared__ memory)∅;                     ▷ set the bucket B to emptyset;
14:
15:    while B ≠ emptyset do
16:        Req ← ∅;                                  ▷ the vertices used to be in current bucket;
17:        while B(id) ≠ emptyset do
18:            u = u0;
19:            while u<|V| do
20:                if u ∈ B(id) then
21:                    Req ← Req ∪ {u};
22:                    for each (u, v, w) ∈ E do
23:                        if w ≤ Δ then             ▷ the light edge;
24:                            atomicMin(&predist(v), dist(u) + w);
25:                        end if
26:                    end for
27:                end if
28:                u ← (u + offset);
29:            end while
30:
31:            __syncthreads();
32:            u ← u0;
33:            while u<|V| do
34:                if predist(u)<dist(u) then
35:                    dist(u) = predist(u);
36:                    moveutoB(dist(u)/Δ)
37:                end if
38:                u ← (u + offset);
39:            end while
40:        end while
41:
42:        __syncthreads();
43:        u = u0;
44:        while u<|V| do
45:            if u ∈ Req then
```

```
46:                 for each (u, v, w) ∈ E do
47:                     if w>Δ then                                   ▷ the heavy edge;
48:                         atomicMin(&predist(v), dist(u) + w);
49:                     end if
50:                 end for
51:             end if
52:             u ← (u + offset);
53:         end while
54:
55:         __syncthreads();
56:         u ← u0;
57:         while u<|V| do
58:             if predist(u)<dist(u) then
59:                 dist(u) = predist(u);
60:                 moveutoB(dist(u)/Δ)
61:             end if
62:             u ← (u + offset);
63:         end while
64:
65:         id ← (id + 1);                                            ▷ goto next bucket;
66:     end while
67: end function
68:
69: initial(s, V);
70:
71: host_to_device(dist), host_to_device(G(V, E));                     ▷ copy the dist and
    G(V, E)frommainmemorytoGPUmemory;
72:
73: delta_steppingCudaFunc();                                          ▷ call the CUDA kernal;
74: device_to_host(dist);                                             ▷ copy the dist back;
75:
76: return result
```