Algorithm 1 spfa CPU 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: function initial(s, V)
2:     for each v ∈ V do
3:         dist(v) ← +∞;                          ▷ initialize dist to positive infinity;
4:     end for
5:     dist(s) ← 0;                                ▷ set the source distence to 0;
6:     Q ← Queue;                                  ▷ set Q as a empty queue;
7:     Q ← {s};                                    ▷ put the source to the queue;
8: end function
9:
10: initial(s, V);
11:
12: while Q is not empty do
13:     p ← vertex in Q head;                       ▷ vertex p is in the head of Q;
14:     remove p from Q;
15:     for each (p, v, w) ∈ E do                   ▷ vertex p has a edge to vertex v with the weight w;
16:         if dist(v)>dist(p) + w then
17:             dist(v) ← dist(p) + w;
18:             if p ∉ Q then
19:                 Q ← {v} ∪ Q;                    ▷ put the vertex v to Q tail;
20:             end if
21:         end if
22:     end for
23: end while
24:
25: return result
```