Algorithm 1 edge 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: end function
7:
8: initial(s, V);
9:
10: flag ← 1;                                      ▷ the break flag;
11: i ← 0;
12: while i<|V| do
13:     if flag == 0 then                          ▷ no vertex update it's dist;
14:         break;
15:     end if
16:     flag ← 0;                                  ▷ set the flag t0 0;
17:     for each (u, v, w) ∈ E do
18:         if dist(v)>dist(u) + w then
19:             dist(v) ← dist(u) + w;             ▷ update the dist(v);
20:             flag ← 1;
21:         end if
22:     end for
23: end while
24:
25: return result
```