Algorithm 1 dijkstra CPU SSSP

```
Input: $G(V, E)$, source vertex $s$;
Output: $dist(v)$, $(v \in V)$, the weight of the shortest path from $s$ to $v$;
function $initial(s, V)$
    for each $v \in V$ do
        $dist(v) \leftarrow +\infty$;            ▷ initialize $dist$ to positive infinity;
    end for
    $dist(s) \leftarrow 0$;                      ▷ set the source distence to 0;
    $Q \leftarrow PriorityQueue$;                ▷ the shorter the dist, the more priority;
    $Q \leftarrow \{s\}$;                        ▷ put the source to the priority queue;
end function

$initial(s, V)$;

while $Q$ is not empty do
    $p \leftarrow$ vertex in Q top;              ▷ vertex $p$ has the min $dist(v)$ in $Q$;
    remove $p$ from $Q$;
    for each $(p, v, w) \in E$ do                ▷ vertex $p$ has a edge to vertex $v$ with the weight $w$;
        if $dist(v) > dist(p) + w$ then
            $dist(v) \leftarrow dist(p) + w$;
            $Q \leftarrow \{v\} \cup Q$;         ▷ put the vertex $v$ to $Q$;
        end if
    end for
end while

return $result$
```