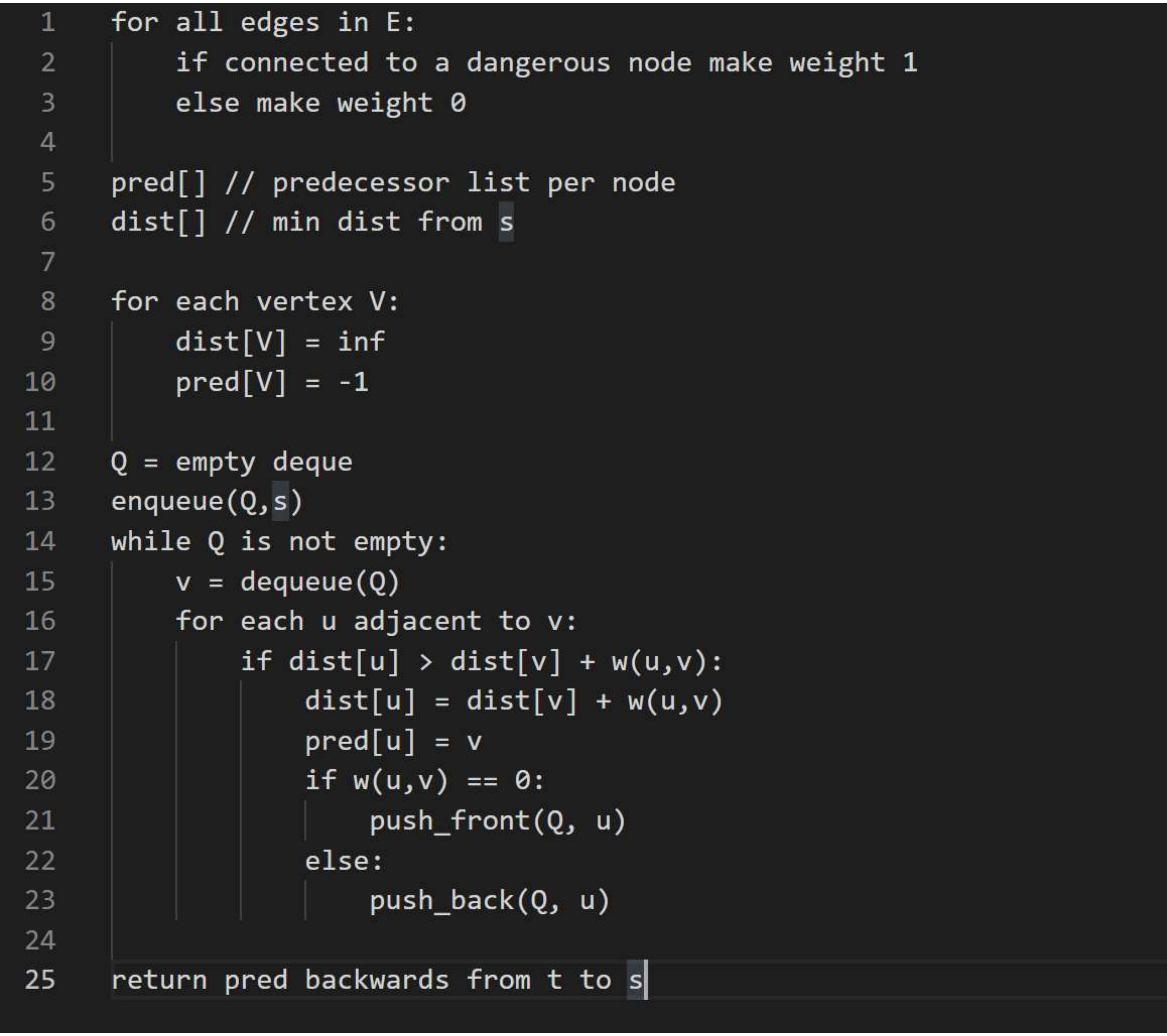

```
for all edges in E:
    if connected to a dangerous node make weight 1
    else make weight 0

pred[] // predecessor list per node
dist[] // min dist from s

for each vertex V:
    dist[V] = inf
    pred[V] = -1

Q = empty deque
enqueue(Q,s)
while Q is not empty:
    v = dequeue(Q)
    for each u adjacent to v:
        if dist[u] > dist[v] + w(u,v):
            dist[u] = dist[v] + w(u,v)
            pred[u] = v
            if w(u,v) == 0:
                push_front(Q, u)
            else:
                push_back(Q, u)

return pred backwards from t to s
```