WEIGHTED-MAJORITY(E, T, n, γ)

```
for i = 1 to n
    w_i^(1) = 1                                   // trust each expert equally
for t = 1 to T
    each expert E_i ∈ E makes a prediction q_i^(t)
    U = {E_i : q_i^(t) = 1}                        // experts who predicted 1
    upweight^(t) = Σ_{i:E_i∈U} w_i^(t)             // sum of weights of who predicted 1
    D = {E_i : q_i^(t) = 0}                        // experts who predicted 0
    downweight^(t) = Σ_{i:E_i∈D} w_i^(t)           // sum of weights of who predicted 0
    if upweight^(t) ≥ downweight^(t)
        p^(t) = 1                                  // algorithm predicts 1
    else p^(t) = 0                                 // algorithm predicts 0
    outcome o^(t) is revealed
    // If p^(t) ≠ o^(t), the algorithm made a mistake.
    for i = 1 to n
        if q_i^(t) ≠ o^(t)                         // if expert E^(i) made a mistake ...
            w_i^(t+1) = (1 − γ)w_i^(t)             // ... then decrease that expert's weight
        else w_i^(t+1) = w_i^(t)
    return p^(t)
```