$$PR(v) := \alpha\left(\frac{1}{N}\right) + (1-\alpha)\sum_{u \in inNeighbors(v)} \frac{PR(u)}{outDegree(u)}$$

Ⓧ $$divClause(v) = \frac{PR(v)}{outDegree(v)}$$

Ⓨ $$PR'(v) := \sum_{u \in inNeighbor(v)} divClause(u)$$

Ⓩ $$PR(v) := \alpha\left(\frac{1}{N}\right) + (1-\alpha)PR'(v)$$

```
function PageRankIteration(graphCSR,PR)
    for v ∈ graphCSR do
      divClause←{0}
  ① divClause[v]←PR[v]/outDegree[v]
    end for
    function PageRankPrime(PR', divClause)
    for v ∈ graphCSR do
      PR←{0}
  ⑤ PRnext[v]←a(1/N)+(1-a)PR'[v]
    end for
  ⑥ PageRank-Swap(PR,PRnext)
```

```
function PageRankPrime(PRprime, divClause)
   for v ∈ graphCSR do
     Sum←{0}
   ② for u ∈ inNeighbor[v] do
       ③Sum←Sum+divClause[u]
     end for
   ④ PRprime[v]←Sum
   end for
```

④ ② ① Sequential access, each cacheline has high reuse

⑤ ③ ALU Operations, Running Sum

③ Random Access Fetch Vertex Property Data