Algorithm 1: RandomWalk

Input: Graph $G(V, E)$, vertex v_i, walk length t

Output: A vertex chosen at random from the neighbors of vertex v_k

indexes = $Neighbors\ of\ vertex\ v_i\ based\ on\ lengths\ t$;
i = $random(indexes)$;
return i

Algorithm 2: DeepWalk(G,w,d,γ,t)

Input: graph G(V,E)
window size w
embedding size d
walks per vertex γ
walk length t

Output: matrix of vertex representations $\Phi \in \mathbb{R}^{|V| \times d}$

Initialization: Sample Φ from $\mathcal{U}^{|V| \times d}$
Build a binary Tree T from V
for $i = 0$ **to** γ **do**
    $\mathcal{O}$ = Shuffle(V)
    foreach $v_i \in \mathcal{O}$ **do**
        $\mathcal{W}_{v_i} = RandomWalk(G, v_i, t)$
        SkipGram($\Phi, \mathcal{W}_{v_i}, w$)
    end
end

Algorithm 3: SkipGram($\Phi, \mathcal{W}_{v_i}, w$)

for $v_j \in \mathcal{W}_{v_i}$ **do**
    for $u \in \mathcal{W}_{v_i}[j - w : j + w]$ **do**
        $J(\Phi) = -\log \Pr(u_k | \Phi(v_j))$
        $\Phi = \Phi - \alpha * \frac{\partial J}{\partial \Phi}$
    end
end