Algorithm 1 iterativeSnapshot(R-tree R_P for indoor POIs, A1R-tree R_O for OTT, time point t, integer k)

initialize a hash table $flows : \{POI\} \rightarrow [0, +\infty]$
for each POI p **do** $flows[p] \leftarrow 0$
LeafEntrySet $les \leftarrow R_O$.PointQuery(t)
for each leaf entry $le \in les$ **do**
    $o \leftarrow le.Ptr_c.objectID$
    $ring_1 \leftarrow Ring(le.Ptr_p.deviceID, V_{max} \cdot (t - le.Ptr_p.t_e))$
    if $le.Ptr_p.t_e < t < le.Ptr_c.t_s$ **then** ▷ The object is in an inactive state
        $ring_2 \leftarrow Ring(le.Ptr_c.deviceID, V_{max} \cdot (le.Ptr_c.t_s - t))$
        $UR(o, t) \leftarrow ring_1 \cap ring_2$
    else ▷ The object is in an active state
        $UR(o, t) \leftarrow ring_1 \cap le.Ptr_c.deviceID.Range$
    $ps \leftarrow R_P.IntersectionQuery(UR(o, t))$
    for each POI $p \in ps$ **do**
        $flows[p] \leftarrow flows[p] + \frac{Area(UR(o,t) \cap p)}{Area(p)}$
return the top-k from $flows.keys$ with the highest values