

**Algorithm 4** *iterativeInterval*(R-tree  $R_P$  for indoor POIs, A1R-tree  $R_O$  for  $OTT$ , time interval  $[t_s, t_e]$ , integer  $k$ )

---

- 1: initialize a hash table  $flows : \{POI\} \rightarrow [0, +\infty]$
  - 2: **for** each POI  $p$  **do**  $flows[p] \leftarrow 0$
  - 3: LeafEntrySet  $les \leftarrow R_O.RangeQuery([t_s, t_e])$
  - 4: initialize a hash table  $H$
  - 5: **for** each leaf entry  $le \in les$  **do**
  - 6:     append  $le.S$  to  $H[le.objectID]$
  - 7: **for** each key  $objectID \in H.keys$  **do**
  - 8:     get  $(rd_s, \dots, rd_e)$  from  $H[objectID]$
  - 9:     calculate  $UR(objectID, [t_s, t_e])$  from  $(rd_s, \dots, rd_e)$
  - 10:      $ps \leftarrow R_P.IntersectionQuery(UR(objectID, [t_s, t_e]))$
  - 11:     **for** each POI  $p \in ps$  **do**
  - 12:          $flows[p] \leftarrow flows[p] + \frac{Area(UR(o, [t_s, t_e]) \cap p)}{Area(p)}$
  - 13: **return** the top- $k$  from  $flows.keys$  with the highest values
-