Algorithm 2: Filter out disks which are subsets

```
1  Algorithm FilterCandidates(B)
       Input: B: active boxes of timestamp t_i, sorted by x-axis values
       Output: C: final set of disks for timestamp t_i
2      C ← ∅
3      for j ← 0 to j ⩽ |B| do
4          for k ← j + 1 to k ⩽ |B| do
5              if IntersectsWith(B[j], B[k]) then
6                  foreach c ∈ B[j].disks do
7                      C ← InsertDisk (C, c)
8              else // No intersection.
9                  break

10 Procedure InsertDisk(C, c)
       Input: C: set of disks, c: new disk
11     foreach d ∈ C do
12         if c.sign ∧ d.sign = c.sign && dist(c, d) ⩽ ε then // c can be a subset of d
13             if d ∩ c = c then // Remove chance of false-positive
14                 return C // No need to insert c
15     else if c.sign ∧ d.sign = d.sign then // d can be a subset of c
16         if c ∩ d = d then // Remove chance of false-positive
17             C ← C \ d // Remove d
18     return C ∪ c
```