Algorithm 1: Naïve Single-Pass Insertion-Streaming Algorithm for Neighbourhood Detection

```
require: Stream {(s_0, t_0) ... (s_n, t_n)}, degree bound d, precision bound c
N ← {{}} {neighbourhoods}
for i = 0 ... n do
    append t_i to N[s_i]
    if size(N[s_i]) = d/c then return (s_i, N[s_i]) ;
return FAIL
```

Algorithm 2: One-pass c-approximation Insertion-Only Streaming Algorithm for `Neighbourhood Detection`

```
require: Space s, degree bound d.
s ← ⌈ln(n) · n^(1/c)⌉
for i ∈ [0, c − 1] in parallel do
    (a_i, S_i) ← Deg-Res-Sampling(max{1, i · d/c}, d/c, s)
return Uniform random neighbourhood (a_i, S_i) from the successful runs
```

Algorithm 3: One-pass c-approximation Insertion-Deletion Streaming Algorithm for `Neighbourhood Detection`. (Vertex Sampling)

```
require: Space s, degree bound d.
Let x = max{n/c, √n}
Sample a uniform random subset A' ⊆ A of size 10 x ln n of vertices.
for a ∈ A' do
    Run 10 (d/c) ln n l_0-samplers on the set of edges incident to a.
return Any neghbourhood of size d/c among the stored edges, if there is one.
```

Algorithm 4: One-pass c-approximation Insertion-Deletion Streaming Algorithm for `Neighbourhood Detection`. (Edge Sampling)

require: Space s, degree bound d.

Let $x = \max\left\{\frac{n}{c}, \sqrt{n}\right\}$

Run $20\frac{nd}{c}\left(\frac{1}{x} + \frac{1}{c}\right)\ln(n)$ l_0 samplers on the stream, in parallel. Each producing a uniform random edge.

return *Any neghbourhood of size* $\frac{d}{c}$ *among the stored edges, if there is one.*