Algorithm 1: PartitionGraph

```
input  : graph, initial partitioning
output: new partitioning
// split
partition ← partition with largest average entropy
foreach node in partition do
    if average entropy of partition reduces without node then
        assign node to new partition
// update
foreach node in sourceNodes do
    assign node to partition with minimal cost
// merge
foreach partition pair ($p1$, $p2$) do
    if total encoding cost decreases when $p1$ merged with $p2$ then
        merge ($p1$, $p2$)
```

Algorithm 2: SegmentGraphStream

```
input  : segment, new graph
newGraphCost ← totalCost(newGraph);
segmentCost ← totalCost(segment);
unionCost ← totalCost(segment ∪ newGraph);
if unionCost- segmentCost <newGraphCost then
    segment ← segment ∪ newGraph
```