Algorithm 1 H1 Greedy Elimination

```
function method($motifs, n$)
    $supernodes \leftarrow \varnothing$
    for $i \leftarrow 1$ to $n$ do
        $shuffled\_motifs \leftarrow$ random_shuffle($motifs$)
        $disjoint\_motifs \leftarrow \varnothing$
        for each $motif \in shuffled\_motifs$ do
            if $motif$ has not overlaps in $disjoint\_motifs$ then
                Add $motif$ to $disjoint\_motifs$
            end if
        end for
        if $disjoint\_motifs$ contains more motifs than $supernodes$ then
            $supernodes \leftarrow disjoint\_motifs$
        end if
    end for
    return $supernodes$
end function
```

Algorithm 2 H2 Ramsey

```
function method($motifs, k$)
    $candidates \leftarrow \varnothing$
    if $motifs$ contains overlaps then
        $flag \leftarrow True$
    else
        $flag \leftarrow False$
    end if
    while $flag$ do
        $flag \leftarrow False$
        Divide $motifs$ in $n$ samples of size at most $k$, call them $subsets$
        for $subset \in subsets$ do
            Build an overlap graph $g$
            Perform the Ramsey algorithm on $g$ and save the maximum independent set in $mis_g$
            Move all motifs from $mis_g$ to $candidates$
        end for
        if $candidates$ contains overlaps then
            $motifs \leftarrow candidates$
            $flag \leftarrow True$
        end if
    end while
    return $candidates$
end function
```

Algorithm 3 H3 Ranked Elimination

```
function method($g, motifs$)
    $PotentialSupernodes \leftarrow \varnothing$
    Compute the degree for each $motif \in motifs$ and build a list of ordered motifs called $MotifDegree$
    for $node \in g$ do
        Move the $motif$ that contains $node$ with the lowest degree from $MotifDegree$ to $PotentialSupernodes$
    end for
    for each pair $m'$, $m''$ of overlapping motifs in $PotentialSupernodes$ do
        Remove the motif with the highest degree between $m'$, $m''$ from $PotentialSupernodes$. Remove a random one if they have the same degree.
    end for
    return $PotentialSupernodes$
end function
```

Algorithm 4 H4 Repeated Ranked Elimination

```
function method($g, motifs$)
    $Supernodes \leftarrow \varnothing$
    while $motifs \neq \varnothing$ do
        $PotentialSupernodes \leftarrow \varnothing$
        Compute the degree for each $motif \in motifs$ and build a list of ordered motifs called $MotifDegree$
        for $node \in g$ do
            Move the $motif$ that contains $node$ with the lowest degree from $MotifDegree$ to $PotentialSupernodes$
        end for
        for each pair $m'$, $m''$ of overlapping motifs do
            Remove the motif with the highest degree between $m'$, $m''$ from $PotentialSupernodes$. Remove a random one if they have the same degree.
        end for
        Consider the list $N_{orphan}$ of all nodes that are not covered by a motif in $PotentialSupernodes$
        Filter $motifs$ keeping only motifs that apply to $N_{orphan}$
        Move all motifs from $PotentialSupernodes$ to $Supernodes$
    end while
    return $Supernodes$
end function
```

Algorithm 5 H5 Sampled Ranked Elimination

```
function method($motifs, k$)
    $candidates \leftarrow \varnothing$
    if $motifs$ contains overlaps then
        $flag \leftarrow True$
    else
        $flag \leftarrow False$
    end if
    while $flag$ do
        $flag \leftarrow False$
        Divide $motifs$ in $n$ samples of size at most $k$, call them $subsets$
        for $subset \in subsets$ do
            Build an overlap graph $g$
            Remove motifs that overlap from $g$ starting with the motifs with the higher degree than other motifs and save the surviving in $mis_g$
            Move all motifs in $mis_g$ to $candidates$
        end for
        if $candidates$ contains overlaps then
            $motifs \leftarrow candidates$
            $flag \leftarrow True$
        end if
    end while
    return $candidates$
end function
```