

```
// store candidate nodes for every unrolled iteration
1: C = array(PAT_SIZE, UNROLL, MAX_DEGREE);
2: Csize = array(PAT_SIZE, UNROLL);
3: iter = array(PAT_SIZE);
// iterate for unrolled iterations
4: uiter = array(PAT_SIZE);
5: l = 0;
6: while (true) {
    if (l < Q.size) {
        // if in the first of the unrolled iterations
        // and if candidate set is empty
        if (uiter[l] == 0 && Csize[l][0] == 0) {
            // extend subgraphs for all unrolled iterations
            getCandidates(G, Q, l, C, Csize, UNROLL);
            // if no subgraph can be extended
            if (l == 0 && Csize[0][0] == 0) {
                // try to steal from other warps
                if (!local_stea�()) {
                    if (!global_stea�()) { break; }}}
            iter[l] = 0; uiter[l] = 0; }
        // if there are more unrolled iterations
        if (uiter[l] < UNROLL) {
            // and if there are unexplored nodes in current
            // unrolled iteration, go to next level
            if (iter[l] < Csize[l][uiter[l]]) { l++; }
            else {
                // if all candidates are explored in current
                // unrolled iteration, go to next unrolled iteration
                Csize[l][uiter[l]] = 0;
                iter[l] = 0;
                uiter[l]++; } }
        else {
            // if all unrolled iterations have been executed
            // reset unroll iterate
            uiter[l] = 0;
            // and backtrack to previous level
            if (l > 0) { l--; iter[l] += UNROLL; } } }
    else {
        for (i = 0; i < UNROLL; i++) Output(C, i, iter);
        l--; } }
```