Segment Match Refinement

The function refine($\mathcal{M}$) will make a k-partite graph with node sets $V^0 \ldots V^{k-1}$ and edges E, which defines boundaries of the minimal resolved refinement of $\mathcal{M}$.

```
def cut(a):
    L = {M^k = (S^i_{u,v}, S^j_{x,y}) | u < a < v ∨ x < a < y}
    for M^k ∈ L:
        b = match position of a in sequence l given by M^k
        if b ∉ V^l:
            insert b into V^l
            insert edge (k,a,b) into E
            cut(b)

def refine(M):
    V^i = boundary positions of segments in sequence i
    for M^k = (S^i_{u,v}, S^j_{x,y}) ∈ M:
        insert edge (k,u,x) and (k,v,y) into E
        for boundary position w of M^k:
            cut(w)
    lexicographically order E
```

A pair of consecutive edges (k, u, v) and (l, x, y) in E, with $k = l$ defines a new segment match $M = \left(S^i_{u,x}, S^j_{v,y}\right)$ between sequences i and j. If you did not note the sequences the positions are from, you just have to look between which sequences the segment match i was defined.