

## 1. Computing Vertex Peel values:

### 1.a. Vanilla version (You have already done.)

### 1.b. Computing Vertex Peel values as well as connected components together

IN: ASCII format edge list `<src>, <tgt>\n`

OUT1: array of peel values for each vertex.

OUT2: array of connected component index for each vertex.

Note: when you compute peel value, you iterate all edges. You can insert a union-find while iterating all edges.

### 1.c. Binary version of 1.a.

IN: binary format edge list, where every 8 Bytes represent an edge, and first 4 Bytes are source, and the last 4 Bytes are target. They are coded as an `uint32` (little-endian).

Namely, `<src><tgt>`.

OUT: binary format list, where every 8 Bytes represent a vertex and its peel value, the first 4 Bytes are vertex index, and the last 4 Bytes are peel value. They are coded as an `uint32` (little-endian). Namely, `<vertex><peel>`

### 1.d. Binary version of 1.b.

IN: binary format edge list, where every 8 Bytes represent an edge, and first 4 Bytes are source, and the last 4 Bytes are target. They are coded as an `uint32` (little-endian).

Namely, `<src><tgt>`.

OUT: binary format list, where every 12 Bytes represent a vertex and its peel value, the first 4 Bytes are vertex index, the middle 4 Bytes are peel value and the last 4 Bytes are connected component index. They are coded as an `uint32` (little-endian). Namely, `<vertex><peel><cc>`

## 2. Computing peel connected component of each vertex using disjoint sets (You have already done.)

## 3. Computing meta node

### 3.a. Vanilla version (You have already done.)

### 3.b. Computing 2 as well as 3 together

IN: array of peel values for each vertex.

OUT1: ASCII list `vertex, peel, peelCC`

OUT2: ASCII list `peelCCIdx, |V|, |E|`

Note: In task 2 you iterate all edges, and you can keep two arrays of counters to record the vertex and edge size of each peel connected component.

## 4. Computing Meta Edges for the Meta graph

We expect the processing rate to reach 2 million edges per second. As we discussed before, you may check peel value rather than peelCC index to speed up.