# Combinatorial Optimization - Programming Exercise 1

March 2020

The implementation will be done in C++.
We can divide the implementation in 3 sub-tasks:

1. Implement `BIPARTITE PERFECT MATCHING`: This will give us the basic structure for the full algorithm. The changes would be mainly in the `Graph`(Given to us - there is no problem in updating it) and `Tree`(Implemented by us). The rest would be additions.
2. Implement `PERFECT MATCHING`: Add *shrink* and *unshrink* (Add modify the classes to handle pseudo-nodes).
3. Implement `MAXIMUM CARDINALITY MATCHING`: Make modifications to the input graph and repeatedly call `PERFECT MATCHING`.

**Tasks**

☐ Implement `BIPARTITE PERFECT MATCHING` (Lucas)

☐ Think on the *shrink* and *unshrink* steps [more detailed *pseudo-code*?] (Kim)

Next meeting planned on Friday 27th - 4pm.