Overview
During the last years, a wide range of huge networks has been made available to researchers. The discovery of natural groups, a task called graph clustering, in such datasets is a challenge arising in many applications such as the analysis of neural, social, and communication networks.

The general idea behind evolutionary algorithms (EA) is to use mechanisms which are highly inspired by biological evolution such as selection, mutation, recombination and survival of the fittest. An EA starts with a population of individuals (in our case partitions of the graph) and evolves the population into different populations over several rounds. In each round, the EA uses a selection rule based on the fitness of the individuals (in our case the edge cut) of the population to select good individuals and combine them to obtain improved offspring.

The main task of this thesis is to define and implement an evolutionary algorithm to tackle the problem. The includes the implementation of a number of heuristic algorithms to combine and mutate given input clusterings. The basic idea is to take two clusterings, compute an overlap and improve it using local search. Roughly speaking, an overlap clustering puts two vertices in the same block if the pair is in the same block in both input clusterings.

\[
\begin{align*}
\forall i \in [1, \ell], v, w \in V : c_{P_i}(v) &= c_{P_i}(w) \Rightarrow c_{\tilde{P}}(v) &= c_{\tilde{P}}(w) \\
\exists i \in [1, \ell], v, w \in V : c_{P_i}(v) &\neq c_{P_i}(w) \Rightarrow c_{\tilde{P}}(v) &\neq c_{\tilde{P}}(w)
\end{align*}
\]

The choice of local search determines the objective that the evolutionary algorithm will optimize. This project has a high chance to be highly successful project. The implementation can be done using data structures provided within the Karlsruhe High Quality Graph Partitioning framework. If the project turns out to be successful, it will be integrated into a coarse grained distributed evolutionary framework.

Requirements
- Interest in algorithms and data structures
- Excellent programming skills in C++
- Ability to think for yourself

INFO: More topics available such as algorithm configuration.