1 Clique Problem

- *Algorithm:* Finding maximum clique in a given graph (algo. 1)
- *Input:* An undirected graph with a set of vertices V and a set of edges E.
- *Complexity:* complexity of the algorithm, e.g. $\mathcal{O}(n)$
- *Data structure compatibility:* Undirected Graphs.
- *Common applications:* DNA computing, sociology studies, data mining.

Clique Problem

The clique problem is to find the maximum clique in a given undirected graph. A clique is defined as all nodes are connected to each other through an edge, namely a complete subgraph.

Description

Suppose we have an undirected graph with a set of vertices and a set of edges, we wish to find a fully connected subgraph with maximum nodes, which is known as a *maximum complete subgraph*. The input shall be the undirected graph.

We will introduce a basic algorithm used to solve the clique problem, called **B&B framework**. (Branch and Bound)

References

- http://www.cs.ecu.edu/karl/6420/spr16/Notes/NPcomplete/clique.html
- Wikipedia is not acceptable if this is the unique reference
- Reference some books, or published articles
- Use reliable websites (no blog allowed) that are not likely to disappear any time soon

Algorithm 1: name

```
Input  : A graph $G$
Output: Clique $C$
Function Main($G$):
    /* C* is the final clique of our algorithm, which will be used in later algorithms
    */.
    C* ← $\varnothing$;
    Clique($C*$,$\varnothing$, $V$);
    return $C*$;
end
Function Clique(set $C*$, set $C$, set $P$):
                                                    /* $|C|$ means the cardinality of $C$ */
    if $|C| > |C*|$ then
        C* ← C;
    end if
    if $|C| + |P| > |C*|$ then
        for $p$ in $P$ do
            P ← P \ {$p$};
            C' ← C ∪ {$p$};
            P ← P ∩ vertices adjacent to $p$;
        end for
    end if
end
return
```