Graph Algorithms

TD3 : Complexity

Throughout this TD, given a graph G, n is its number of vertices, and m its number of edges.

1 Algorithmic complexity

1. Give an explicit implementation of a bucket queue so that one can compute a degeneracy ordering of a given graph G in time $O(m)$.
2. Write an algorithm in pseudo-code that computes a $\Delta(G)$-colouring of a connected graph G that is neither complete nor an odd cycle. It should have complexity $O(m)$.

2 NP-completeness

In this exercise, we study the NP-completeness of the INDEPENDENTSET problem.

1. **Reduction from SAT**. Let $C_1, \ldots, C_r$ be the clauses in an instance X of SAT. Construct a clique of size $|C_i|$ for every clause C_i, and label its vertices with the literals that appear in C_i. Add an edge between every pair of vertices labelled with opposite literals x and $\overline{x}$. This returns a graph G_X. Show that the instance X is satisfiable if and only if $\alpha(G_X) \geq r$.
2. **Reduction from COLOR**. Let G be a graph. Let $k \cdot G$ be the graph obtained by replacing each vertex $v \in V(G)$ in G with a clique of size k (denote its vertices $v_1, \ldots, v_k$), and each edge $uv \in E(G)$ with the complete matching $u_1v_1, \ldots, u_kv_k$. Show that $\chi(G) \leq k$ if and only if $\alpha(k \cdot G) \geq |V(G)|$.

3 VERTEXCOVER is FPT

The problem VERTEXCOVER consists in deciding if a graph G contains a vertex cover, that is a set of vertices X such that each edge $e \in E(G)$ has an extremity in X, of size at most k.

Algorithm 1: VertexCover

```
Data: G: graph, k: integer
Result: decides whether G has a vertex cover of size ≤ k
if E(G) = ∅ then
 | return True
end
if k = 0 then
 | return False
end
uv ← an edge from E(G)
G1 ← G \ u
G2 ← G \ v
return VertexCover(G1,k-1) ∨ VertexCover(G2,k-1)
```

1. Prove that the algorithm is correct.
2. Compute its complexity. In what parameter is VERTEXCOVER FPT?