# Exercise 7.9

maandag 2 oktober 2023 17:16

Consider the following algorithm outline:

1. Re-interpret the undirected graph $G$ as a directed graph $G^*$, where the edges are changed as to point from a vertex with lower index to a vertex with higher index; i.e. if $G^* = (V, E^*)$, then $E^* = \{(v_i, v_j) \in E | i < j\}$. This does not cost extra I/Os, since we can simply ignore any adjacency list entries where $i \geq j$.
2. Define a function $f$ on the nodes of $G^*$ as follows:
   a. $f(v_i) = 1$ if $i = 1$ *(note: this assumes the vertex indices range from* $1$ *to* $n$*)*
   b. If $i > 1$, then $f(v_i) = \begin{cases} 1 & if \ \sum_{v_{in} \in N_{in}(v_i)} f(v_{in}) = \sum_{v \in \{v_j \in V | 1 \leq j < i\}} f(v) \\ 0 & otherwise \end{cases}$
3. This function is not local, given that the sum $\sum_{v \in \{v_j \in V | 1 \leq j < i\}} f(v)$ requires elements from nodes other than the in-neighbors as well. Therefore, theorem 7.2 cannot be applied directly. However, it is easy to see how this issue can be worked around: given that this sum effectively sums all the $f$-values obtained so far, we can let the algorithm keep a separate variable which simply stores the sum's value and increments it as needed. *(This costs no more than 1 block of internal memory, and we assume this memory is available.)*
4. It is now straightforward to see that the set of vertices $C = \{v_i | f(v_i) = 1\}$ form a clique; any vertex added to this set will have the property that, whenever it is added, *the number of vertices in its in-neighbors which is in the set is equal to the number of vertices which is in the set*. This clique is also maximal; any vertex which did not satisfy this property at the point where it was considered cannot be part of the clique (given that there was at least one node which was added to the clique, but which was not in the in-neighbors of the non-added vertex).
5. Given that the function $f$ is local *(after addressing the caveat under point 3)*, and that this function can be computed in $O(SORT(1 + |N_{in}(v_i)|))$ I/Os, it follows from theorem 7.2 that this algorithm (outline) can be used to compute a maximal clique in $O(SORT(|V| + |E|))$ I/Os.