Assignment 2 – applied assignment
COMP6741: Parameterized and Exact Computation

2015, Semester 2

Assignment 2 is based on group work. Each group consists of 2–3 students and selects to do either the applied assignment or the theoretical assignment. Each group selects a name for the group and one corresponding author, who will be responsible for the submission of the assignment. Register your group on WebCMS3 with type “Assignment2” by 02 September 2015. Post on the Forum if you are looking for a group to join, or if your group is short of members.

Please cite any works (articles, Wikipedia entries, lecture notes, etc.) that inspired your solutions.

If you have questions about this assignment, please post them to the Forum.

Due date. This assignment is due on Thursday, 08 October 2015, at 23.59 AEST. Submitting x days after the deadline, with x > 0, reduces the grade by 20 · x per cent.

Submission. Submit a TAR(.GZ) or ZIP archive with the following files

- a report in PDF format (the first page should contain the names and Student IDs of each group member), and
- all source files of the implementations (choose any common programming language), and
- two shell scripts mycompile.sh and myrun.sh such that mycompile.sh compiles the code (in case you use an interpreted language, this shell script might simply do nothing), and myrun.sh executes one of the algorithms on a randomly generated instance.

Submit this archive using the command
give cs6741 a2 <myarchive>
from the CSE network, or use the new WebCMS3 frontend for give.

Applied assignment

The aim of this assignment is to implement and compare (FPT) algorithms for Feedback Vertex Set.

A feedback vertex set of a graph \( G = (V, E) \) is a set \( S \subseteq V \) such that \( G - S \) is acyclic.

<table>
<thead>
<tr>
<th>Feedback Vertex Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input: ( G = (V, E) ) and an integer ( k ).</td>
</tr>
<tr>
<td>Parameter: ( k )</td>
</tr>
<tr>
<td>Question: Does ( G ) have a feedback vertex set of size ( k )?</td>
</tr>
</tbody>
</table>

1. Implement the Feedback Vertex Set algorithm from Section 4.3.1 of [1]. [10 points]
2. Implement another Feedback Vertex Set algorithm of your choice. [10 points]
3. In your report, describe your implementations. What data structures do you use for graphs. How do you ensure that small modifications to the graph do not take too much time? How do you pass the instance in the recursive calls of the branching algorithms? Is it necessary to create a copy of the graph for each recursive call? [25 points]

4. Design and implement a method to generate instances for Feedback Vertex Set. Describe this method in your report. Can you make sure to select a value for $k$ that equals the size of a smallest feedback vertex set or the size of a smallest feedback vertex set minus 1. [25 points]

5. Run these algorithms on generated instances with a time-out of 10 minutes. Describe your environment (processor, programming language, how do you measure execution time, etc.) and save the instances in files so that your results will be reproducible. Describe your choice for what instances to generate. How many instances do you generate? Graph the results (number of solved instances, average time, etc.) according to the value of the parameter and/or the number of vertices. Interpret the results. Do they match your expectations? Do Yes- or No-instances seem harder to solve? Which instances seem particularly hard to solve? [30 points]

**Bonus Question**

- Show that, parameterized by $|R| + \Delta^{-}(D)$, Defendable Extension is not FPT, under standard complexity-theoretic assumptions, such as P $\neq$ NP or FPT $\neq$ W[1]. Here, $\Delta^{-}(D) = \max_{v \in V} \{|N_D(v)|\}$ is the largest in-degree of a vertex of $D$. See the theoretical assignment for the definition of the Defendable Extension problem. [10 points]

**References**