CS170 – Project
Deliverable 1: Design Doc

Xu-Bin Kuang 3032136979
Keyan Nasseri 3031901744
Jason Kafil 3031880515

In the building of our inputs, we wanted to direct our optimum solution in an unconventional direction and the following details highlight our best effort in achieving that.

The Prison Bus:
We wanted to create an instance such that the optimum solution would include a bus filled of students that are not friends with each other, but if put on any other bus, would prevent a naïve algorithm from achieving a good solution. We introduced *s* nodes that we classified as 'Prisoners'. They are friends with everyone else but involved in several specified rowdy crowds that prevent the optimum solution from including them in any bus. This initial condition covers the backbone of our algorithm. Failure to detect this would lead to a sub-optimal solution.

This requires the optimum solution to have a bus with an abnormally low score.

Other Busses:
Within each bus, we introduced s/3 'Jock' nodes. These are nodes that are popular with s/2 people in their respective optimal bus but popular with s people in some other bus. However, placing this Jock in any bus other than its designated bus would result in a lower score overall as Jocks form rowdy crowds with certain jocks from other busses.

This traps algorithms that greedily maximize number of friends. It requires an algorithm that consider combinations of nodes, instead of individual nodes to avoid this. Prisoners are pairwise rowdy with all Jocks, thus any decision to misplace a prisoner would result in both Jocks and the Prisoners getting kicked off the bus.

Pseudo-Popular:
Pseudo-Popular nodes are nodes that are the most strongly connected in the graph after prisoners. They form rowdy crowds with all but one jock on the bus, forcing most algorithms to choose between a jock or a pseudo-popular node. However, pseudo-popular nodes form rowdy crowds with themselves and will not, under any circumstance, be able to make the trip.

This traps algorithms that do not consider sole rowdy crowds. Given their strong connectivity, we have written this with the assumption that some algorithms will optimize for connectivity and overlook the single rowdy crowds.