York University CSE 3101 March 7, 2012

Homework Assignment #8
Due: March 19, 11:30 a.m.

1. Alviine organizes timetables at a fitness centre. There are n different activities $a_1, \ldots, a_n$ to schedule. There are k timeslots to use for the activities. Alviine has a list of members who are interested in each activity. Ideally, activities should be scheduled so that, if any member is interested in two activities, those two activities are scheduled in different timeslots (so that the member can attend both of them). We call such a schedule of all the activities a *legal* schedule. Alviine wants to figure out whether a *legal* schedule exists.

(a) First, consider the case where $k = 2$. Below is a greedy algorithm to solve the problem. We use an array $T[1..n]$ to store the timeslots assigned to activites: $T[i]$ stores the timeslot assigned to job a_i. The timeslots are numbered 1 and 2.

```
let $A = \{a_1, \ldots, a_n\}$
initialize all entries of $T$ to 0
for $i = 1..n$
    invariant: $A$ is the set of all activities that have not been assigned timeslots
    if $\exists$ two activities $a_i \in A$ and $a_j \notin A$ such that some member is interested in both then
        $T[i] = 3 - T[j]$ % i.e., assign $a_i$ the opposite timeslot of $a_j$
        remove $a_i$ from $A$
    else
        remove any activity $a_i$ from $A$
        $T[i] = 1$
    end if
end for
if any member is interested in any two activities $a_i$ and $a_j$ such that $T[i] = T[j]$ then
    output "impossible"
else output "possible"
end if
```

Let $T_k[i]$ be the value stored in $T[i]$ after k iterations of the loop. Carefully prove the following claims:

Claim 1: If a legal schedule exists, then, for all k, there is a legal schedule $T^*[1..n]$ such that for all i, if $T_k[i] > 0$ then $T_k[i] = T^*[i]$.

Claim 2: If a legal schedule exists, then the algorithm outputs "possible".

Claim 3: If the algorithm outputs "possible", then a legal schedule exists.

(b) Now consider the case $k = 3$. Suggest a reasonable greedy algorithm that uses a fairly simple decision procedure for scheduling each activity. Then, show your algorithm is wrong by giving an input that causes your algorithm to output an incorrect result.