CS 4820, Spring 2019 **Homework 7, Problem 2**
Name: Kevin Klaben NetID: kek228

(2) *(10 points)* You are organizing a carnival for the town of Ithaca. There are various artists, arriving from all over the globe, to perform in various shows as part of the carnival. A total of n artists are participating from a total of m countries. Further, there are a total of r shows. The following are the constraints on organizing the carnival:

1. Artist ℓ, subject to her skills, is only capable of participating in a subset of the shows given by a list L_ℓ.
2. The i'th show requires exactly n_i artists to perform.
3. Each show can have at most one artist from each country.
4. So as to be fair to all artists, each artist can participate in at most r' shows.

As part of the input, you will receive integers $n, m, r, r', n_1, \ldots, n_r$ and the capability list $L_1, \ldots, L_n$ of the artists. Your task is to design an efficient algorithm to determine if you can organize the carnival subject to the above restrictions. If indeed it is possible, you have to output the allocation of artists to shows.

Algorithm:

```
carnival(n,m,r,r',$n_1$,...,$n_r$,$L_1$,...,$L_n$)
initialize a graph $G = (A \cup C \cup S \cup SOURCE \cup SINK, E)$ where E={}, A={$a_1$,...,$a_n$}, and C={$c_{1,1}$,...,$c_{m,r}$} for all 1..m and all 1..r, and S={$s_1$,...,$s_r$}
for each artist ability list $L_i$ do
  for each show $s_j$ in $L_i$ do
    add an edge to E from $a_i$ to $c_{k,j}$ with weight=1 where k is the country $a_i$ is from
  end for
end for

for each show $s_i$ in i=1..r do
  for each k in 1..m do
    add an edge to E from $c_{k,i}$ to $s_i$ with weight=1
  end for
end for

for each i in 1..n do
  add an edge to E from SOURCE to $a_i$ with weight=r'
end for
for each i in 1..r do
  add an edge to E from $s_i$ to SINK with weight=$n_i$
```

```
end for
value=Compute a maximum flow f of G using the Edmonds-Karp algorithm
int sum=0
for each i in 1..r do
  sum=sum+$n_i$
end for
if sum!=value then
  return "Not possible"
end if
Construct the residual graph $G_f$.
Define the set $E_{A,B} =$ e=(u,v) E: uA and vC output={}
for each e in $E_{A,B}$ do
  if e which is from $a_i$ to $c_{k,j}$ has weight zero in $G_f$ then
    add pair ($a_i$,$s_j$) to output
  end if
end forreturn ("possible",output)
```

Runtime Analysis:
The first nested for-loop takes time O(nr) as there are n artist lists and r shows. The second set of nested for-loops takes time O(rm) as there are r shows and m countries being looped over. The third for-loop takes O(n) time and the fourth and fifth each take O(r) time. The Edmonds-Karp algorithm takes O(($edges^2$*vertices)). As there are n artist vertices, r show vertices and mr country vertices, there are a total of n+mr+r vertices. m is bound by n and thus we can say there are O(n+nr+r) vertices. There are a total of n source to artist edges, maximum of nr artist to country edges, nr country to show edges, and r show edges, thus the total number of edges is O(n + nr + nr + r) edges. Thus overall, the Edmonds-Karp algorithm would take time O($O(n + nr + r)^2$O(n+nr+nr+r)) time which simplifies to O(n^3*r^3). The final output for-loop must iterate over nr edges and for each edge must check if that edge in G_f has zero weight which could take potentially O(nr) time iterating and finding this edge in G_f again. Thus the final for-loop could take O(n^2*r^2). Thus the overall runtime is bound by Edmonds-Karp and will have a time of O(n^3*r^3).

Proof of Correctness:
Claim:
The algorithm outputs "Not possible" if no possible scheduling of artists to shows exists and outputs "possible" and one possible assignment of artists to shows such that all constraints are satisfied.

Subclaim:
If there is no possible assignment of performers to shows such that all constraints are satisfied, the algorithm returns "not possible".
Proof: Suppose for the sake of contradiction that the algorithm outputs "possible" and some assignment of performers to shows. Well, this assignment must violate at least one constraint of the problem as this assignment cannot exist according to our assumption. The algorithm will only output "possible" when the max-flow output by the edmonds-karp algorithm is equal to the sum of the maxflows on the show nodes going to the sink. The weights on these edges are equal to the minimum amount for each show, thus if all are fully used then each show has had it's

minimum number of performers met. In order for each show to have met the this flow minimum at least these many countries must have sent artists as a countries edge weight to a show cannot be greater than 1. This also means that each country sends a maximum of one performer to each show thus satisfying this condition. As this country node is only connected to artists via edges of weight 1 which have been placed by reading through the performers ability list. As only edges have weights if the performer is able to perform at that show this list is not violated. Only one artist will have a flow of one leading to the country and only artists who are able to perform will be potentially selected to provide this flow. The artists are linked to the source via edges with weight equal to the maximum performances. As the flow over this edge cannot exceed the edge weight, performers cannot perform in more than the maximum number of shows. Thus, the condition of each performer not exceeding the max r' will hold for this final scheduling. As all conditions of the problem are satisfied by this matching of artists to shows, this is a valid matching. This is a contradiction as we assumed that there was not a valid matching in this problem. Thus, the claim holds.

Subclaim: The algorithm outputs ”possible” and one possible assignment of artists to shows such that all constraints are satisfied.
Proof: For the sake of contradiction, suppose that there exists a possible valid assignment of artists to shows satisfying all constraint but the algorithm outputs ”not possible”. The only way for ”not possible” to be output is if the sum of the weights on the edges going from the shows to the sink is not equal to the Max-Flow. This means that the max-flow from edmonds-karp in the graph that was setup must be less than the sum of the weights of these final edges. This means that at least one show must have not had enough performers to meet the minumum specified by the problem. In order to meat the condition that at most one artist from each country may perform at a show, the edges connecting countries to shows all have weight 1. The sum of the flows over these edges must have been less than the minimum for at least one show. The flows over these edges cannot be satisfied only when an artists had already met the maximum number of performances allowed, or there is not an artists from that country with the ability to perform in such a show. In the case that there are not enough countries with performers with abilities to perform in a show, then there does not exist a possible matching and thiscontradicts the assumption. In the case that an artist has already been met their maximum number of performances, Edmonds-Karp will check all possible cases such that the fow can be increased. If the flow to the show can be increased by shifting around this performers and other performers flows it will be found by the Edmonds-Karp algorithm and will be shifted. If this shift is possible then the edmonds-Karp would have not returned such a low value thus this must not be possible. As there is no way to shift flows to satisfy this show without breaking another shows minimum, there does not exist a possible matching for this problem which satisfies all conditions. This again is a contradiction with our assumption and thus the claim holds.

Proof of overall claim: In both the case that an assignment does not exist, the algorithm always outputs ”Not possible” (proven by the subclaim) and in the case that there does exist such an assignment, then the algorithm always outputs ”possible” and a valid assignment (proven by the other subclaim). As both of these cases holds, we can conclude that the overall algorithm always outputs the correct answer for this problem and thus the claim holds.