HW 9 Yizhou Liu CS 577 liu773 @ wisc.edu

1、 a) CounterExample is given as following:

$n = 6$, $k = 2$, C would be

$$\begin{vmatrix} 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{vmatrix}$$

To be specific, disc 1 can be placed on 3, 4, 5, 6

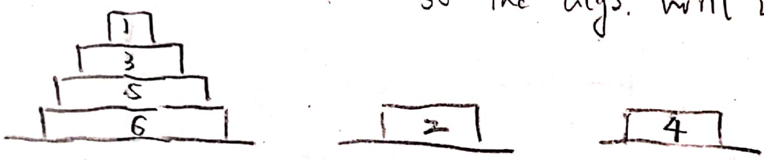
disc 2 can be placed on 5, 6.

disc 3 can be placed on 4, 5, 6.

disc 5 can be placed on 6.

disc 4 & 6 cannot be placed on any discs.

By the greedy algo, we need 3 pegs to place these discs, so the algo. will return "No".



However, we can place the discs as followings so that we only 2 pegs to place all these discs, so the correct answer is "Yes".

Algo:

We first construct a graph from C by creating edges of cap. 1 between all nodes satisfying $c_{ij} = 1$.

For every vertex in the graph, we define it into 2 unconnected vertices, v_{in} and v_{out} (i.e., their demand are $1, -1$, respectively). By the definition above, we get edge (v_{in}, v_{out}) has demand 1, edge (v_{out}, v_{in}) has demand -1.

Then, connect all v_{in} to vertex (demand $-l$)

connect all v_{out} to vertex (demand l)

connect each vertex (negative demand d_v) to meta-source s with an edge (cap. $= -d_v$).

Connect each vertex (positive demand d_v) to meta-sink t with an edge (cap $= d_v$).

Perform Ford-Fulkerson algo. to get max s-t flow f

if $\sum_{v: d_v > 0} d_v = f$, return "Yes".

else, return "No"

Proof of Correctness

We only need to show that this problem can be reduced to airline scheduling problem. For the placing discs problem, since ① each vertex (disc) needs to be placed on peg and ② each vertex can be put on certain other discs, easy to see that performing a Ford-Fulkerson algo. can get the min number of matches needed. And this is exactly the same as our goal to find a feasible arrangement of n discs to k pegs. Other parts of this algo's correctness follows correctness of airline scheduling problem.

Time Complexity

$m = \#$ of edges $= \frac{n(n-1)}{2} + 2n$ (max) $= O(n^2)$

max cap. of an edge $= l \le n$

Then, time complexity is $O(mnl) = O(n^4)$.