HW 9 Yizhou Liu CS 577 liu773@wisc.edu

1、 a) CounterExample is given as following:

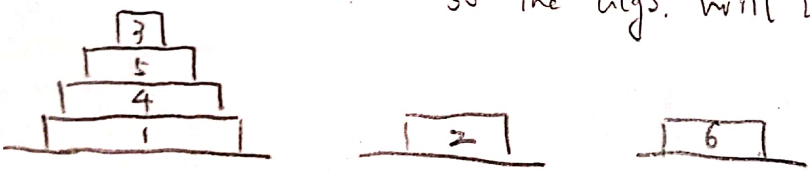
$n = 6$, $k = 2$, C would be

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

To be specific, disc 1 & 2 cannot be placed on any disc.

disc 3 can be placed on disc 1, 2, 5.
disc 4 can be placed on disc 1.
disc 5 can be placed on disc 1, 2, 4
disc 6 can be placed on disc 1, 4.

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 divide 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"

Program 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

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

$$\text{max cap. of an edge} = l \leq n$$

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