HW9

2018/12/3

1. **Counterexample:**

N = 6, k = 2, C Matrix:

A description of C:

Discs1 is the biggest and all discs can be placed on it.

Discs2 can only be placed on Discs1.

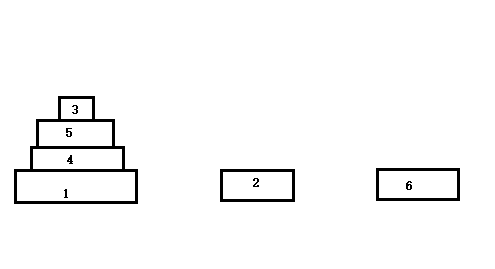
Discs3 can be placed on Discs1, 2, 4, 5, 6.

Discs4 can only be placed on Discs1.

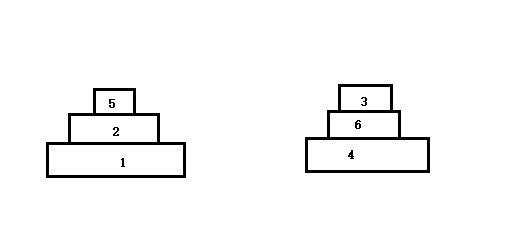
Discs5 can be placed on Discs1, 2, 4.

Discs6 can be placed on Discs1, 4.

According to the greedy algorithm, the number of the pegs it needs is 3:



But in fact, the correct number of pegs is 2 and the situation is “Yes”



1. **Design Algorithm:**

We can treat the problem as maximum bipartite matching. Firstly, we construct a bipartite graph G. G has 2n vertices, one on the left and one on the right for each discs. There is an edge from discs-i on the left to discs-jon the right if discs-i can be placed on discs-j. Then try to find the maximum matching M of G. If the maximum matching of G is (n – k) or more, these discs can be placed on these k pegs, if not, these discs cannot be placed on k pegs and should print “No”.

**Correctness:**

If we want to place these n discs on k pegs, we should try to place at least (n – k) discs on other discs. Say in other words, there are at least (n – k) matchings in the bipartite graph. So if the discs can be placed on these pegs, the maximum matching of G is (n – k) or more. The algorithm is correct.

**Running Time:**

As we all know that maximum bipartite matching runs in time O(|V| |E|), so in this problem maximum bipartite matching runs in time O(n^3). Meanwhile the construction of bipartite graph G costs time O(n^2). So the total time-complexity is O(n^3).