

Ground Rules

- The homework is worth 10 points (out of a total of 100 points you can accumulate in this course).
- The homework is to be done and submitted individually. You may discuss homework problems with *at most one* other person from either section (and mention this person's name on your submission), but you must write up solutions on your own.
- You are not allowed to consult any material outside of assigned textbooks and material the instructors post on the course websites. In particular, consulting the internet will be considered plagiarism and penalized appropriately.
- The homework is due at 11:59 PM on the due date. No extensions to the due date will be given under any circumstances.
- Homework must be submitted electronically on canvas. We will not accept paper submissions. Canvas accepts files in pdf/doc/jpeg formats.

Problem (page limit: 3 pages)

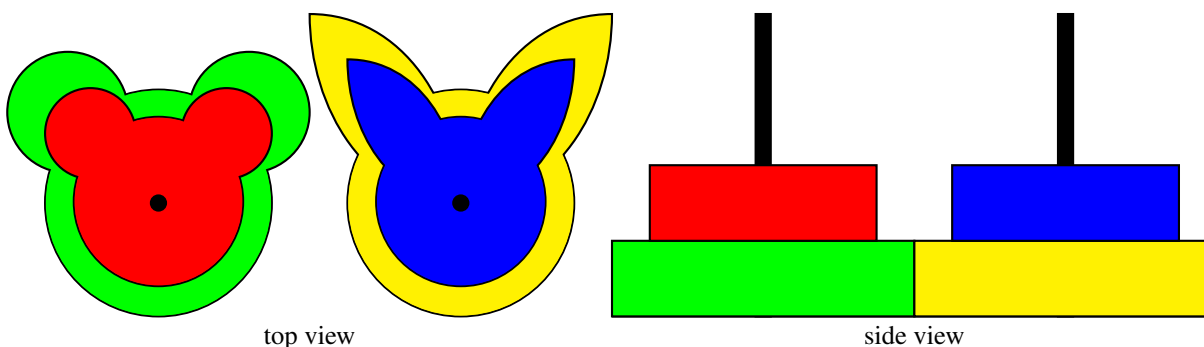
Children's toymaker AlgosRUs is creating a stacking puzzle where n oddly shaped discs need to be placed on k pegs, and has employed you to help with the task. Each disc can either be placed directly on a peg or must be placed on top of another disc that is strictly larger than it in all directions. Your task is to decide whether n given shapes can be assembled into k "stacks" to be placed on the pegs.

As input to the problem, you are given an $n \times n$ matrix C , where for all $i, j \in [n]$, $C_{ij} = 1$ if disc i can be placed on disc j and $C_{ij} = 0$ otherwise. You can assume that the given matrix represents a collection of shapes that are physically possible. Given n , C , and k as input, your goal is to design an algorithm that will output "Yes" if a feasible arrangement of the n discs on k pegs exists, and "No" otherwise.

For example, suppose that $n = 4$ and the matrix C is given by:

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

Then the discs can be arranged on two pegs as shown in the picture below. However, the discs cannot be arranged on a single peg.



- (a) Show that the following greedy algorithm does NOT solve this problem correctly by providing a counterexample. Make sure you include the matrix C and either a description of the shapes consistent with C or a diagram showing the shapes.

Algorithm 1: Greedy

Input: The list of discs D , the compatibility matrix C , and the number of pegs k

```
1 for  $i \leftarrow 1$  to  $k$  do
2   Find the largest number of discs from the list  $D$  that can be put on one peg, call it  $n_i$ , and remove these
   discs from the list  $D$ ;
3 if  $n_1 + n_2 + \dots + n_k = n$  then
4   return YES
5 else
6   return NO
```

Line 2 can be done with a dynamic programming algorithm, but you do not need to specify the details. If your counterexample relies on a particular tie-breaking rule (i.e., a rule that specifies which discs to remove in line 2 when there are multiple longest sequence of discs that can be put on one peg), please specify it. However, to obtain full marks for this part, your counterexample should not rely on the tie-breaking rule.

- (b) Design an algorithm that solves this problem and has runtime polynomial in n . Provide a brief proof of correctness, as well as an analysis of the running time of your algorithm.

Hint: try to reduce this problem to maximum flow, minimum cut, or some other application of network flow discussed in the lectures. You do not need to present an algorithm for solving the problem you reduce to, but make sure you explain how the reduction is made and argue its correctness.