Algorithms – CS-37000
The "Greedy matching" problem

A *matching* in a graph $G = (V, E)$ is a set $M \subseteq E$ of pairwise disjoint edges. The *size* of a matching is the number of edges in M. The *matching number* $\nu(G)$ is the maximum size of a matching in G. For instance, $\nu(K_n) = \lfloor n/2 \rfloor$ and $\nu(K_{r,s}) = \min\{r, s\}$ (verify these to make sure you understand the definition!).

A matching M is *maximal* if it cannot be extended, i. e., if there is no matching that properly contains M.

A matching M is *maximum* if $|M| = \nu(G)$.

Note that every maximum matching is maximal but not conversely (verify!).

We wish to estimate $\nu(G)$ using a greedy approach. It will turn out that the worst error we can make in doing so is a factor of 2.

We assume $V = \{1, 2, \ldots, n\}$ and E is given as a list $e_1, e_2, \ldots, e_m$.

The *greedy matching algorithm* is described by the following pseudocode:

Initialize: $M := \emptyset$
for $i = 1$ to m **do**
    if e_i does not intersect any edge in M **then** add e_i to M
end(for)
return M

It should be clear that the algorithm returns a matching (why?).

Problem. (a) Let M be the matching returned by the greedy matching algorithm. Prove:

(a1) M is a maximal matching.

(a2)

$$|M| \le \nu(G) \le 2|M|. \tag{1}$$

(b) Prove that the upper bound is tight in the following sense:

$(\forall k \ge 0)(\exists G = (V, E)$ and an ordering of $E)(\nu(G) = 2k$ and $|M| = k)$.