CSE 421 HW1

Rohan Mukherjee

April 2, 2024

1. (a) We give the following counter-example:

$$
\begin{array}{lcl}
c_1 : a_1 > a_2 > a_3 & \qquad & a_1 : c_3 > c_2 > c_1 \\[2em]
c_2 : a_3 > a_1 > a_2 & & a_2 : c_2 > c_1 > c_3 \\[2em]
c_3 : a_2 > a_3 > a_1 & \text{———} & a_3 : c_1 > c_3 > c_2
\end{array}
$$

We see that each company is paired with their second favorite applicant and each applicant their second favorite company, and this this is indeed a perfect matching. We need only check stability, and we can do so by just checking it on the applicants. Since there is only company a_1 likes more than its current pair, we only need to check that that company does not also prefer a_1. Indeed, c_3's least favorite applicant is a_1 and that is below its current pair of a_3. The same pattern holds for each of the other a_i's, completing the counter-example.

(b) We give the (very complicated) counter-example:

$$
\begin{array}{lcl}
c_1 : a_1 > a_4 > a_3 > a_2 & \qquad & a_1 : c_4 > c_3 > c_2 > c_1 \\[2em]
c_2 : a_1 > a_4 > a_3 > a_2 & & a_2 : c_4 > c_3 > c_2 > c_1 \\[2em]
c_3 : a_1 > a_4 > a_3 > a_2 & & a_3 : c_4 > c_3 > c_2 > c_1 \\[2em]
c_4 : a_4 > a_1 > a_2 > a_3 & & a_4 : c_1 > c_2 > c_3 > c_4
\end{array}
$$

If c_1 gets its favorite, c_4 must also get it's favorite because it cannot have either a_2 or a_3 or else it would deviate with a_1. Then if we have $c_2 \leftrightarrow a_2$ and $c_3 \leftrightarrow a_3$ or vice versa, both will want to deviate with a_1.

If c_2 gets its favorite, once again c_4 must be with it's favorite otherwise a_1 will prefer c_4. Then in any arrangement of c_1 and c_3, they both prefer a_4 over their current match and a_4 prefers them over their current match. The case c_3 is exactly the same and is thus skipped.

If c_4 gets its favorite, c_1 must get its favorite otherwise c_1 and a_4 will deviate. Then in any assignment of c_2 and c_3, they both prefer a_4 over their current matches and a_4 prefers them as well.

Thus there is no stable matching where a company gets its favorite in this instance.

2. Suppose instead that at least two companies c_1, c_2 get their last choice (where there are n companies c_i $1 \leq i \leq n$ and n applicants a_i $1 \leq i \leq n$). Our first observation is the last choice of c_1 is distinct from c_2 (since we have a perfect matching), and WLOG we shall call them a_1 and a_2 respectively. We observe that for c_1 to be rejected by it's first $n-1$ choices, those $n-1$ choices must currently have their final company currently married with them. This is because (1) for an applicant to replace / reject a company, it needs to either have gotten a new company with a better priority, or already have a company with better priority, in any case it will have a company. Next we observe (2) that it must be the case that a_1 is free, by our last observation we have used up the rest of the $n-1$ companies. Then the algorithm will assign c_1 to a_1 and then immediately terminate, since no other company is free. In particular, it cannot be the case that the second company was rejected by a_1, otherwise a_1 by observation (1) would not be free, a contradiction. Thus, $n-1$ of the companies make at most $n-1$ offerings, and the last company, the only one who potentially gets it's last choice, makes n offerings, giving us a total of

$$(n-1)(n-1) + n = (n-1)(n-1) + n - 1 + 1 = n(n-1) + 1$$

3. We proceed by induction on k. For $k = 0$, we have $2^0 = 1$ numbers, say x. The condition $\sum_i x_i = 1$ just says that $x = 1$. In this case, $\sum_i x_i^2 = \sum_i 1^2 = 1 \geq \frac{1}{2^0}$, completing the base case. In general, assume that for every list of 2^k real numbers $z_1, \ldots, z_{2^k}$ satisfying $\sum_{i=1}^{2^k} z_i = 1$ we have that

$$\sum_i z_i^2 \geq \frac{1}{2^k}$$

And let $x_1, \ldots, x_{2^{k+1}}$ be a list of 2^{k+1} numbers. Consider the list of 2^k numbers $y_i = x_{2i-1} + x_{2i}$ for $1 \leq i \leq 2^k$ (i.e, $y_1 = x_1 + x_2$, $y_2 = x_3 + x_4$ and so on). Clearly,

$$\sum_{i=1}^{2^k} y_i = \sum_{i=1}^{2^{k+1}} x_i = 1$$

Thus we see that

$$\frac{1}{2^k} \leq \sum_{i=1}^{2^k} y_i^2 \leq \sum_{i=1}^{2^k} (2x_{2i-1}^2 + 2x_{2i}^2) = 2 \sum_{i=1}^{2^{k+1}} x_i^2$$

Thus,

$$\sum_i x_i^2 \geq \frac{1}{2^{k+1}}$$

which completes the proof.

4. Our final answer is

$$a \ll h \ll c \ll i \ll g \ll b \ll j \ll e \ll d \ll f$$

Meaning:

$$2^{2\sqrt{\log n}} \ll n^{\frac{1}{\log\log(n)}} \ll \frac{n(\log\log(n))^{99}}{\log(n)^{99}} \ll 2^{\log n - \log\log n}$$
$$\ll \log(n!) \ll 2^{\log\left(n^2\right)} \ll (4^2)^{\log n} \ll 4^{2^{\log n}} \ll n!^2 \ll n^{n \log n}$$