6th tutorial

1. Goldberg's Implementation: Implement Goldberg's algorithm as follows. Let S be a list of vertices with positive surplus, and for each $v \in V$, let $K(v)$ be a list of edges going out of v with positive reserve and going down, i.e. $K(v) = \{vw \mid r(vw) > 0 \wedge h(v) > h(w)\}$.
How to maintain S and $K(v)$'s during each push and each relabel (raising a vertex higher)? What will be the resulting complexity? (Recall from the lecture that there are $\geq 2n^2$ relabels, $\leq nm$ saturated pushes, and $\leq n^2 m$ unsaturated pushes.)

2. Goldberg and Unit Capacities: Analyze Goldberg's algorithm on networks with unit capacities. Will it be faster than other algorithms? Or at least as fast?

3. Goldberg with Highest Vertex: Design an implementation of improved Goldberg's algorithm where we always lift the highest vertex with a positive surplus. The number of unsaturated pushes will be $\mathcal{O}(n^2\sqrt{m})$ (you don't need to prove this), your goal is to design an algorithm matching this time.

4. Matrix Rounding: A matrix $A \in \mathbb{R}_{\geq 0}^{r \times s}$ is given, and you would like to round each of its entries up or down to an integer while preserving row and column sums. Design an algorithm which either gives a solution, or correctly declares the problem infeasible.

5. Goldberg and Heights: What would happen, if we placed the source at height $n-1$, $n-2$ or even $n-3$? Figure out which property (it terminates, gives max flow always, etc.) depends on the height of the source, and what could go wrong.