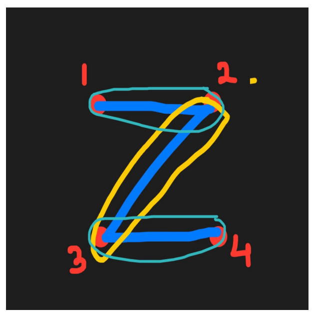
# Linear Programming
# Problem Set 4 – CS 6515/4540 (Fall 2025)

Solutions to problem set 4, Q13.

## 13 Approx Algo

1. We'll prove two things:

   - The greedy algorithm given returns a Maximal matching, as by definition of a maximal matching, there must be no other edge in the graph that can be added to the matching i.e. no other edge exists that is not incident to any vertex in our maximal matching.
   - For any edge in the maxmimal matching, $M$ it can be incident to or correspond to, at max, 2 edges in the maximum matching, $M^*$. This is because for any $e = (u, v) \in M$, $M^*$ must also have edges incident to the vertices $u$, and $v$. Therefore, $|M^*| \leq 2|M| \implies |M| \geq \frac{|M^*|}{2}$.

   In the above undirected graph, with 4 vertices numbered 1 to 4. The yellow set, containing only edge, $(2, 3)$ is a valid maximal matching. And the aqua set, containing edges, $(1, 2), (3, 4)$ is a maximum matching. Here, $|M^*| = 2 = 2 * 1 = 2 * |M|$.

2. Let's sort the items by weight, then $w_1 \geq w_2 \geq \cdots \geq w_n$ be the item sizes, and let $C^*$ denote the minimum possible maximum weight of a bin for placing these $n$ items into $m$ bins.

   Let
   $$T = \sum_{i=1}^{n} w_i.$$

   If we were allowed to place items fractionally into the bins then the optimal value, $C^* = \frac{T}{m}$, but since we're not allowed to that,
   $$C^* \geq \max\left\{\frac{T}{m},\ w_1\right\}.$$

   Now we run the approximation algorithm, and let $C$ be the resulting maximum load. Let $M$ be the bin that attains the weight $C$ in the approx solution, and let $j$ be the last item placed into $M$, with weight $w_j$. When item $j$ was placed into $M$, the weight of $M$ was $L = C - p_j$.

   Since the algorithm always places the next item in the least-weight bin, every bin had weight at least $L$ at that moment. Therefore, the total weight of all bins before placing $j$ was at least $T - w_j$, hence
   $$T - w_j \geq mL = m(C - w_j).$$

Rearranging, we get

$$mC \le T + (m-1)w_j. \tag{1}$$

We now consider two cases.

(a) $w_j \le \frac{1}{2}C^*$: Using the inequalities $T \le mC^*$ and $w_j \le \frac{1}{2}C^*$ in (1), we have

$$mC \le mC^* + (m-1)\frac{C^*}{2} = \frac{3m-1}{2}C^*$$

$$C \le \frac{3m-1}{2m}C^* \le \frac{3}{2}C^*,$$

since $\frac{3m-1}{2m} \le \frac{3}{2}$ for all integers $m \ge 1$.

(b) $w_j > \frac{1}{2}C^*$: Any item with weight greater than $\frac{1}{2}C^*$ must occupy its own bin, because two such items together would exceed $C^*$ invalidating the optimal constraint. Hence there must be at most $m$ items larger than $\frac{1}{2}C^*$.

Since the approx algorithm processes items in non-increasing order, the heavier items are each placed in a distinct bin because each bin initially empty. After these items are placed, all remaining items satisfy $w_i \le \frac{1}{2}C^*$. Placing any of them into any bin can therefore increase that bin's weight by at most $\frac{1}{2}C^*$. We know, $C^* \ge \max\left\{\frac{T}{m},\ w_1\right\}$. Therefore, any bin that already holds a heavy item ends with weight at most

$$w_1 + \frac{1}{2}C^* \le C^* + \frac{1}{2}C^* = \frac{3}{2}C^*.$$

In both cases, we obtain $C \le \frac{3}{2}C^*$. Hence, the greedy algorithm is a 1.5-approximation for the minimum possible maximum bin weight. $\square$