Lemma 1. *Suppose we have $l+1$ lists of positive integers $x_{i,1}, \dots, x_{i,m_i}$ for $i = 1, \dots, l+1$, where the lists length is $m_i \geq 1$ and the $(l+1)^{\text{th}}$ list contains only one element $x_{l+1,1} = 1$. Denote the sum of entries of a list by $y_i = \sum_{j=1}^{m_i} x_{i,j}$ for $i = 1, \dots, l+1$ and $N = \sum_{i=1}^{l+1} \sum_{j=1}^{m_i} x_{i,j} = \sum_{i=1}^{l+1} y_i$ be the total sum of all elements. Further let every element $x_{i,j}$ satisfies $x_{i,j} \leq \sum_{k=i+1}^{l+1} y_k$, then there is for every integer $z \in \{1, \dots, N\}$ a subset Z of the $x_{i,j}$'s which sum to z.*

Proof. Let $z \in \{1, \dots, N\}$ be arbitrary but fixed. For the proof we introduce an algorithm which produces one desired subset Z whose elements sums to z. This algorithm goes through all lists and all entries of the lists in ascending order, i.e. in order $x_{1,1}, \dots, x_{1,m_1}, x_{2,1}, \dots, x_{2,m_2}, \dots, x_{l,m_l}, x_{l+1,1}$, and adds the current element $x_{i,j}$ to Z if the total sum together with $x_{i,j}$ is at most z. Let $n \in \{0, \dots, l+1\}$ denote the number of lists which are processed and let $\sigma(n)$ be the sum of all elements of Z after the algorithm went through n lists. We will show that $\sigma(n)$ is in the interval $\left[z - \sum_{k=n+1}^{l+1} y_k, z\right]$. This will prove this lemma since $\sigma(l+1) \in [z, z] = \{z\}$ is the sum of all elements of Z after the algorithm terminated.

That the temporary sum $\sigma(n)$ is always smaller or equal z for all $n \in \{0, \dots, l+1\}$ is obvious by the construction of the algorithm. Thus it is only to prove the lower bound $z - \sum_{k=n+1}^{l+1} y_k$ for $\sigma(n)$. We do this by induction on the number n of processed lists. Since Z is empty before the algorithm begins and $z \leq N$ it follows straight-forward for $n = 0$ by

$$\sigma(0) = 0 \geq z - N = z - \sum_{k=1}^{l+1} y_k.$$

For the induction step we suppose that the sum $\sigma(n-1)$ after processing $n-1$ lists is greater or equal than $z - \sum_{k=n}^{l+1} y_k$. If the algorithm goes through the n^{th} list their are two cases, either he adds all entries $x_{n,j}$ for $1 \geq j \geq m_n$ to Z or not. If he can add every entry the sum of all elements of Z will increase by the sum of all elements of the n^{th} list such that

$$\sigma(n) = \sigma(n-1) + y_n \geq z - \sum_{k=n}^{l+1} y_k + y_n = z - \sum_{k=n+1}^{l+1} y_k$$

holds true by the induction hypothesis. Otherwise there is an entry $x_{n,j}$ such if it will be added to Z the sum will be greater than z. Therefore the sum after processing the n^{th} list is also greater than z if we add $x_{n,j}$ since the sum of the elements of Z is only increasing. By the assumption on the upper bound on the entries of the n^{th} list this leads to

$$\sigma(n) = \sigma(n) + x_{n,j} - x_{n,j} \geq z - x_{n,j} \geq z - \sum_{k=n+1}^{l+1} y_k.$$

This shows that the sum of all elements of Z after n processed lists $\sigma(n)$ belongs to the interval $\left[z - \sum_{k=n+1}^{l+1} y_k, z\right]$ and therefore it proves the lemma like mentioned before. □