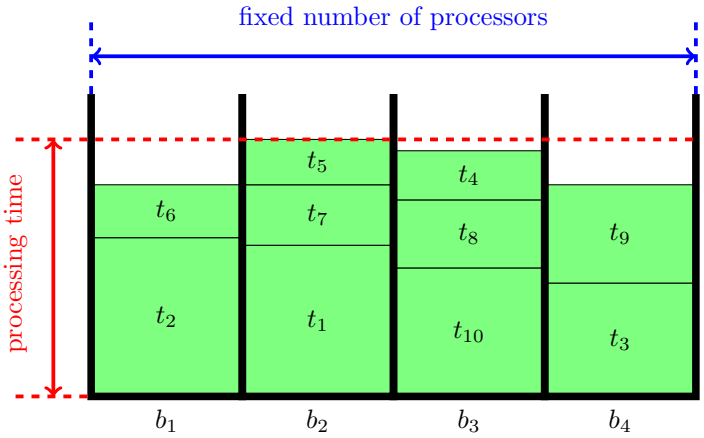
1. In the MULTIPROCESSORSCHEDULING problem, we are given n tasks and p processors. Each task has a time associated with it t_i. The goal of the problem is:

   minimize: $\max\left\{\sum_{i=1}^{n} t_i \delta_{ij} : j \in 1..p\right\}$

   where $\delta_{ij} = 1$ if task i is in bin j and 0 otherwise. In picture form, the goal is to minimize the processing time of the worst processor as shown below:



   (a) Write a greedy algorithm for this problem.

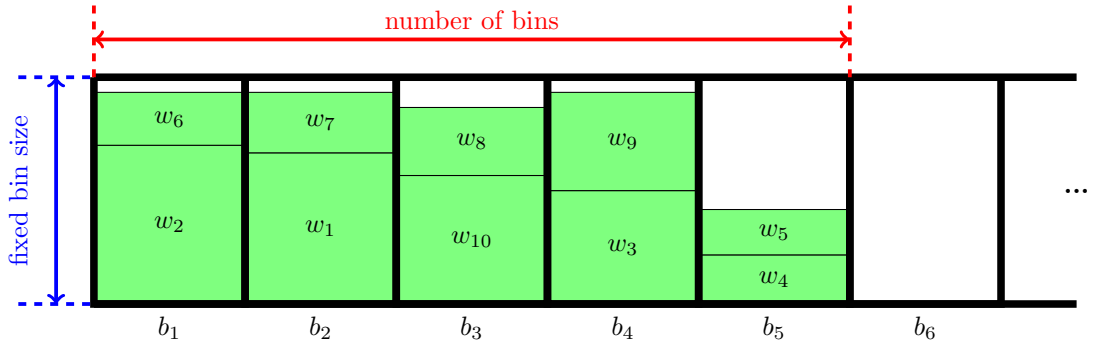
   (b) Is your algorithm optimal? (That is, does it *really* find the minimum solution?) If yes, justify why; otherwise, find a counterexample.

2. In the BinPacking problem, we are given n items and a fixed bin size S. Each item has an associated weight w_1. The goal of the problem is:

$$\text{minimize:} \quad \sum_{i=1}^{n} y_i$$

$$\text{subject to:} \quad \forall j \sum_{i=1}^{n} w_i \delta_{ij} \leq S y_j$$

Where δ_{ij} is defined as above, and y_i equals one if bin i is being used and 0 otherwise. In picture form, the goal is to arrange each item in a bin so that the minimum number of bins is used:



(a) Write a greedy algorithm for this problem.

(b) Is your algorithm optimal? (That is, does it *really* find the minimum solution?) If yes, justify why; otherwise, find a counterexample.