Load Balancing

Consider the problem LOAD BALANCING (or MINIMUM MAKESPAN SCHEDULING): Given $n$ jobs $J_1, \ldots, J_n$ with processing time $p_1, \ldots, p_n$, respectively and $m$ identical machines. Find a schedule of these jobs such that the makespan is minimized.

**Question 1**

Consider the following greedy algorithm.

**Algorithm 1** A greedy algorithm for LOAD BALANCING

1: Order the jobs arbitrarily.
2: for $i = 1$ to $n$ do
3: Assign $J_i$ to the machine with least current load.
4: Update the load.
5: end for
6: return the assignment.

1. Prove that the greedy algorithm gives $(2 - \frac{1}{m})$-approximation and give an tight example for the algorithm.
2. Take a modification on the first step: order the jobs in decreasing order of processing time. Prove that the new algorithm gives $3/2$-approximation

**Hint:** Let $L$ be the makespan and let $M_i$ be a machine with load $L$ (in the output). Consider the last job assign to machine $M_i$.

**Question 2**

Assume that there are $k$ distinct processing times, i.e., $p_i \in \{a_1, \ldots, a_k\}$ where $k$ is a constant. Given $T > 0$. Design a dynamic programming whose running time is $O(n^{2k})$ such that:

- Return No if the minimum makespan is larger than $T$,
- Return Yes and a schedule that has makespan smaller than $T$ otherwise.

**Hint:** Define $M(x_1, \ldots, x_k)$ := the number of machines needed to schedule $x_i$ jobs of processing time $a_i$ (for $1 \leq i \leq k$) such that the makespan is smaller than $T$. 