Exercise 2.3 Prove that for any constant $c > 1$ and for any $n > 3$, there exists an instance $x_{c,n}$ of MINIMUM TRAVELING SALESPERSON with n cities such that *Nearest Neighbor* achieves a solution of measure $m_{NN}(x_{c,n})$ such that $m_{NN}(x_{c,n})/m^*(x_{c,n}) > c$.

Exercise 2.4 (*) Prove that for any constant $c > 1$, there are infinitely many instances $x_{c,n}$ of MINIMUM METRIC TRAVELING SALESPERSON with n cities such that *Nearest Neighbor* achieves a solution of measure $m_{NN}(x_{c,n})$ such that $m_{NN}(x_{c,n})/m^*(x_{c,n}) > c$. (Hint: for any $i > 0$, derive an instance for which *Nearest Neighbor* finds a tour whose length is at least $(i+2)/6$ times the optimal length.)

Exercise 2.5 Consider the MINIMUM SCHEDULING ON IDENTICAL MACHINES problem. Show that the LPT rule provides an optimal solution in the case in which $p_l > m^*(x)/3$.

Exercise 2.6 Prove that the bound of Theorem 2.7 is tight if $p = 2$.

Exercise 2.7 Consider the variation of the LPT rule that assigns the two longest jobs to the first machine and, subsequently, applies the LPT rule to the remaining jobs. Prove that if $p = 2$ then the performance ratio of the solution provided by choosing the best solution between the one given by this variation and the one returned by the original LPT rule is strictly better than the performance ratio of the solution provided by the LPT rule.

Exercise 2.8 Prove that the bound of Theorem 2.9 is tight in the sense that no better multiplicative factor can be obtained if the additive factor is 1.

Exercise 2.9 Show that there exists an instance of MINIMUM BIN PACKING for which the number of bins used by the solution computed by *Best Fit Decreasing* is greater than the number of bins in the solution computed by *First Fit Decreasing*.

Exercise 2.10 Consider the following variant of MINIMUM BIN PACKING: the input instance is defined by a set of n items $\{x_1, x_2, \ldots, x_n\}$ whose sum is at most m. The goal is to maximize the number of items that are packed in m bins of unitary capacity. A sequential algorithm for this problems that is similar to *First Fit* considers items in the given order and tries to pack each item in the first available bin that can include it. If none of the m bins can accomodate item x_i then x_i is not packed. Prove that the above algorithm achieves a solution that packs at least $n/2$ items.

Exercise 2.11 (*) Let us consider the generalization of MINIMUM BIN PACKING to higher dimensions, known as vector packing. In this problem the size of each item x is not a single number but a d-dimensional vector