Assignment 1

Please type your answers using Latex. Please hand in your assignment (by email or by hard copy) before the class of October 21, 2020. You are welcome to discuss among yourselves, but everyone must write down his/her own answers. Please feel free to contact me if you have questions.

Bin-Packing

Given n items whose sizes are $a_1, \cdots a_n \in (0.1]$, we want to put them into bins of size 1 (so a subset S of items can be fit into a bin if and only if $\sum_{i \in S} a_i \leq 1$) , with the objective using as few bins as possible. This is an NP-hard problem. But there is a rather simple greedy algorithm. It works like this:

> You consider items one by one. If you can, put the item into the latest-opened bin (and only this one). If you cannot, open a new bin.

Prove that this algorithm gives a 2-approximation. Also give a tight (or as tightly as possible) example.

Knapsack

In class we present an approximation scheme, which gives us $(1-\epsilon)$-approximation. In practice, there is a much simpler algorithm based on greedy. Let us order the items e by their decreasing "density", namely $\frac{p(e)}{c(e)}$, where $p(e)$ is the profit and $c(e)$ is the size of item e.

Now we construct a set S by adding into S (initially empty) greedily e_1, $e_2, \ldots, e_k$, where e_k is the first item whose addition makes S strictly exceed the knapsack size B. Of course S is not a feasible solution. But prove that there exists a subset of S which is both feasible and gives at least a 1/2-approximation.