**Tutorial: 7**

**Bin Packing Algorithm**

Like the multiple knapsack problem, the bin packing problem also involves packing items into bins. However, the bin packing problem has a different objective: find the fewest bins that will hold all the items.

The following summarizes the differences between the two problems:

* Multiple knapsack problem: Pack a subset of the items into a fixed number of bins, with varying capacities, so that the total value of the packed items is a maximum.
* Bin packing problem: Given as many bins with a common capacity as necessary, find the fewest that will hold all the items. In this problem, the items are not assigned values, because the objective does not involve value.

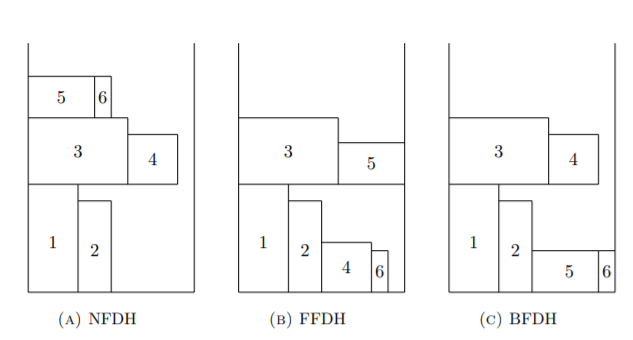
Given n items of different weights and bins each of capacity c, assign each item to a bin such that number of total used bins is minimized. It may be assumed that all items have weights smaller than bin capacity.

Let j be the item we selected, and s the last created level:

**Next-Fit Decreasing Height strategy (NFDH):** Item j is packed left corner of the bin on level s, if it fits. Otherwise, a new level is created, and j is packed left corner of the new level.

**First-Fit Decreasing Height strategy (FFDH):** Item j should be packed on the left corner of the first level where it fits, if any. If no level can accommodate j, a new level is creaated as in NFDH.

**Best-Fit Decreasing Height strategy (BFDH):** Item j is packed left corner of that level where it fits, and the unused horizontal space should be minimized. If there is no space for item j, a new level is created as to make it fit inside the bin.



**Example:**

**Input:** wieght[] = {4, 8, 1, 4, 2, 1}

Bin Capacity c = 10

**Output:** 2

We need minimum 2 bins to accommodate all items

First bin contains {4, 4, 2} and second bin {8, 1, 1}

**Input:** wieght[] = {9, 8, 2, 2, 5, 4}

Bin Capacity c = 10

**Output:** 4

We need minimum 4 bins to accommodate all items.

**Input:** wieght[] = {2, 5, 4, 7, 1, 3, 8};

Bin Capacity c = 10

**Output:** 3

**Lower Bound**  
We can always find a lower bound on minimum number of bins required. The lower bound can be given as:

Min no. of bins >= Ceil ((Total Weight) / (Bin Capacity))

In the above examples, lower bound for first example is “ceil (4 + 8 + 1 + 4 + 2 + 1)/10” = 2 and lower bound in second example is “ceil (9 + 8 + 2 + 2 + 5 + 4)/10” = 3.   
This problem is a NP Hard problem and finding an exact minimum number of bins takes exponential time.

 **Applications**

1. Loading of containers like trucks.
2. Placing data on multiple disks.
3. Job scheduling.
4. Packing advertisements in fixed length radio/TV station breaks.
5. Storing a large collection of music onto tapes/CD’s, etc.

Example1:

Here, items of various weights need to be packed into a set of bins with a common capacity.   
Assuming that there are enough bins to hold all the items. Find the fewest that will suffice using bin packing algorithm.  
Weights are (48, 30, 19, 36, 36, 27, 42, 42, 36, 24, 30) and bin capacity is 100.

Example 2:

There are 50 items packed into a bin. Each item has a value (the number on the item) and a weight (roughly proportional to the area of the item).   
The bin is declared to have a capacity of 850. Find the set of items that will maximize the total value without exceeding the capacity.

Values are (360, 83, 59, 130, 431, 67, 230, 52, 93, 125, 670, 892, 600, 38, 48, 147,  
78, 256, 63, 17, 120, 164, 432, 35, 92, 110, 22, 42, 50, 323, 514, 28,  
87, 73, 78, 15, 26, 78, 210, 36, 85, 189, 274, 43, 33, 10, 19, 389, 276,  
312)

Weights are (7, 0, 30, 22, 80, 94, 11, 81, 70, 64, 59, 18, 0, 36, 3, 8, 15, 42, 9, 0,  
42, 47, 52, 32, 26, 48, 55, 6, 29, 84, 2, 4, 18, 56, 7, 29, 93, 44, 71,  
3, 86, 66, 31, 65, 0, 79, 20, 65, 52, 13).

http://www.cs.ucr.edu/~neal/2006/cs260/piyush.pdf

file:///C:/Users/user/Downloads/MargotvanAken\_Thesis\_censor.pdf

https://www.ics.uci.edu/~goodrich/teach/cs165/notes/BinPacking.pdf

**https://www.youtube.com/watch?v=kiMFyTWqLhc**