KIT205 Data Structures and Algorithms

Week 10 Tutorial

In this tutorial you will be writing code to solve a problem using dynamic programming.

The Knapsack Problem (adapted from 20bits.com)

You are a thief who has broken into a jewellery warehouse. Obviously you can't take everything. In particular, you're constrained to take only what your knapsack can hold — let's say it can only hold W kilograms. You also know the market value for each item of jewellery. Given that you can only carry W kilograms what items should you steal in order to maximize your profit?

Say there are n paintings with weights $w_1, ..., w_n$ and market values $v_1, ..., v_n$. Define A(i,j) as the maximum value that can be attained from considering only the first i items weighing at most j kilograms.

Obviously A(0,j) = 0 and A(i,0) = 0 for any $i \leq n$ and $j \leq W$. If $w_i > j$ then A(i,j) = A(i-1, j) since we cannot include the ith item. If, however, $w_i \leq j$ then A(i,j) then we have a choice: include the ith item or not. If we do not include it then the value will be A(i-1, j). If we do include it, however, the value will be v_i + A(i-1, j - w_i). Which choice should we make? Well, whichever is larger, i.e., the maximum of the two.

Write some code to solve the knapsack problem. All of the code can go in a single file with the main function. You should write a function with the following prototype:

```
float knapsackValue(int w[], int v[], int W, int n);
```

Test your function with the following arrays and various knapsack sizes.

```
int w[10] = {1,1,3,3,2,4,3,6,5,7};
```

```
int v[10] = {100,150,50,25,2,15,1000,25,55,225};
```