CS 218 : Design and Analysis of Algorithms

**Lecture 13: Knapsack**

Lecturer: *Sundar Vishwanathan*
Computer Science & Engineering Indian Institute of Technology, Bombay

# 1 Knapsack

## 1.1 Introduction

As input you are given a set of elements each with a value and a weight and a bound $W$. Among all subsets with weight at most $W$, output one with maximum value. In other words, you have a bag that can carry at most a weight of $W$. You would like to stuff this with elements so that the total value of the elemens stuffed is maximum.

For this problem we will design a procedure that outputs the maximum weight (and not the subset itself.) It is easy to extend the procedure to output the subset as well with some extra book-keeping. We will leave this to you.

The naive method which checks all possible subsets takes exponential time. We give an efficient procedure as long as the individual weights are not too large.

## 1.2 Problem Statement

**Input:** A set of $n$ positive integer weights $w_i$ and values $v(i)$, and a positive integer $W$.
**Output:** A subset of maximum value with weight at most $W$.

## 1.3 Recursion

We begin by defining our procedure and its parameters. $MaxValue(w_1, v_1; w_2, v_2; ...; w_n, v(n); W)$

If the input set of weights is empty, we return 0.

This is also a familiar subset problem-where the output required is a subset of the input. As before the optimal subset may either contain the first element or not. If it contains the first element, the maximum weight can be written as: $v_1 + MaxValue(w_2, v_2, \ldots, w_n, v_n; W - w_1)$. Note that in the recursive call the weight cannot exceed $W - w_1$. If it does not contain the first element, the maximum weight we can get is: $MaxValue(w_2, v_2; \ldots, w_n, v_n; W)$. We pick the maximum of these. What are the distinct procedure calls and what is the generic procedure? Opening this out twice, we can see that the generic procedure can be written as: $MaxValue(w_i, ..., w_n; W')$, for some $W' < W$. Hence number of calls hence is $n \times W$. The recursion for the generic procedure is: $Maxvalue(w_i, v_i, ..., w_n, v_n, W') = \max(v_i + MaxValue(w_{i+1}, ..., v_n, W' - w_i), MaxValue(w_{i+1}, ..., v_n), W')$. There is a technicality to take care of. The first part of the recursion is used provided $w_i \leq W'$, otherwise we recurse only on the second term.

*Exercise.* Write the complete procedure, including the base case and analyse it. Clearly indicate the table that you will use and connect the analysis with the table entries.

### 1.3.1 A comment on the running time

The dependence of the running time of the algorithm on $W$ is not good. If $W$ is an $m$-bit integer, the time taken is proportional to $2^m$ which is exponential in the size of the input. Such algorithms which run in time which is polynomial in the input size provided the weights are

given in unary are called *pseudo-polynomial*. However for this problem, as we shall see later, it is unlikely that better algorithms exist.