Knapsack Problem

By Avanish Shrestha (5919364)

Problem Definition: Knapsack

Given a set of items, each with a weight and a value, determine the number of each item to include in a collection so that the total weight is less than or equal to a given limit and the total value is as large as possible

- 0/1 Knapsack: The number of copies of each kind of item is restricted to zero or one

NP-completeness

A decision problem *c* is NP-complete if:

1. ***c* is in NP**, and
2. Every problem in NP is reducible to *c* in polynomial time

Knapsack ∈ NP

Knapsack is a well known application of dynamic programming; we can solve it using recursion

Let, n = no. of items; w = remaining weight;
W = knapsack capacity

—[1] **m[n, w]** = 0 *if n = 0 or w = 0*

—[2] **m[n, w]** = m[n-1, w] *if* $w_n > w$

—[3] **m[n, w]** = max(m[n-1, w], m[n-1, w-w_n] + v_n)
if $w_n <= w$

```
// Input:
// Values (stored in array v)
// Weights (stored in array w)
// Number of distinct items (n)
// Knapsack capacity (W)

for j from 0 to W do:
    m[0, j] := 0
```

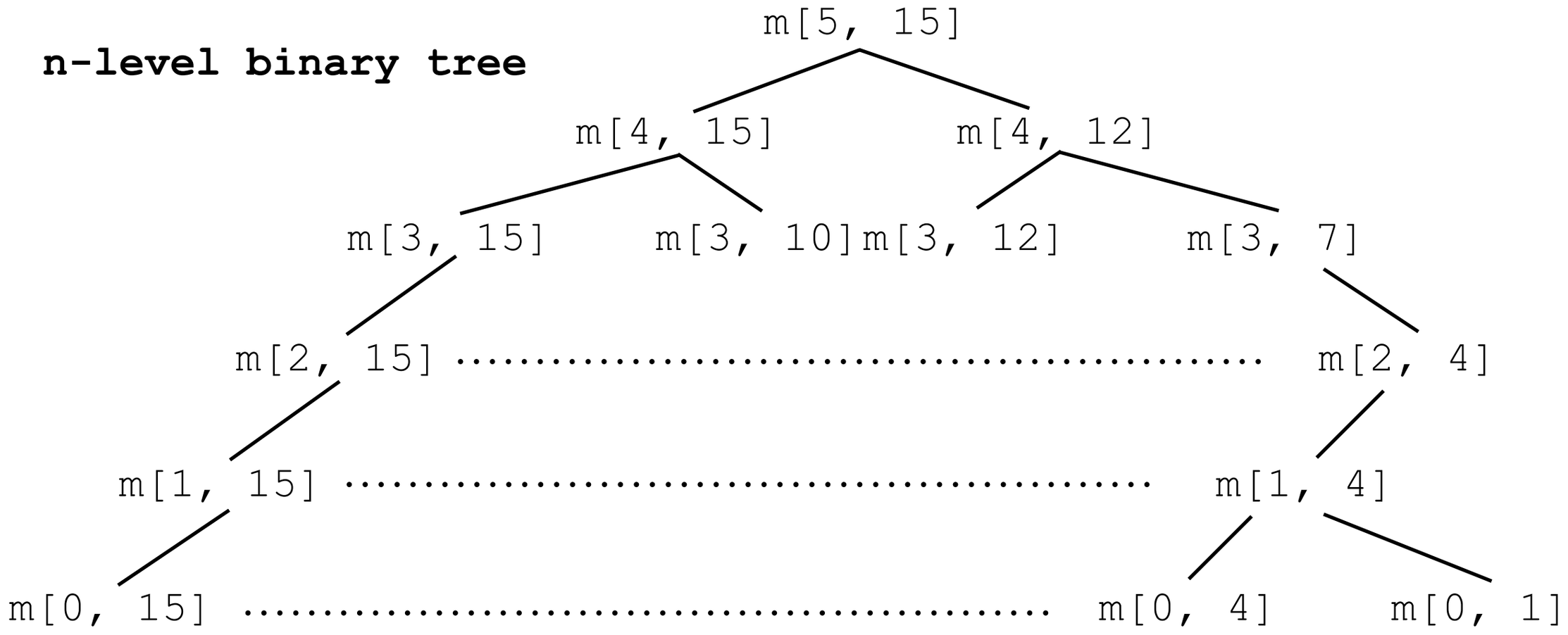
```
for i from 1 to n do:
    for j from 0 to W do:
        if w[i] > j then:
            m[i, j] := m[i-1, j]
        else:
            m[i, j] := max(m[i-1, j], m[i-1, j-w[i]] + v[i])
```

Solution takes O(nW) time.

Example: n = 5; W = 15

Object	1	2	3	4	5
Weight	3	7	3	5	3
Value	20	15	70	60	30

n-level binary tree



2^n nodes, therefore time complexity is $O(2^n)$

Definition of NP-complete

A decision problem *c* is NP-complete if:

1. *c* is in NP, and
2. **Every problem in NP is reducible to c in polynomial time**

Exact-cover

Given a collection *S* of subsets of a set *X*, an **exact cover** is a subcollection *S** of *S* such that each element in *X* is contained in exactly one subset in *S**

Exact-cover

- Let *S = {A, B, C, D, E, F}* be a collection of subsets of a set *X = {1, 2, 3, 4, 5, 6, 7}* such that:
  - *A = {1, 4, 7}*
  - *B = {1, 4}*
  - *C = {4, 5, 7}*
  - *D = {3, 5, 6}*
  - *E = {2, 3, 6, 7}*
  - *F = {2, 7}*
- The subcollection *S* = {B, D, F}* is an exact cover.

Exact-cover $\leq_p$ Knapsack

- Let there be n number of items
- Item i has value v_i and weight w_i
- We are given K and W
- Knapsack: there exists a subset $S \subseteq \{1, 2, \ldots, n\}$ such that $\sum_{i \in S} w_i \leq W$ and $\sum_{i \in S} v_i \geq K$
- Assume $v_i = w_i$ and $W = K$

Exact-cover $\leq_p$ Knapsack

- Knapsack: A subset of $V = \{v_1, v_2, \ldots, v_n\}$ adds up to exactly K

- Let $S = \{S_1, S_2, \ldots, S_x\}$ be the collection of subsets that covers the set V for the exact-cover problem

- We will construct the integers $\{v_1, v_2, \ldots, v_n\}$ and K such that there is a subset P of S with $\sum_{j \in P} v_j = K$ if and only if there is a family of pairwise disjoint sets that covers the whole U.

Exact-cover $\leq_p$ Knapsack

- Each of the *x* subsets in *S* corresponds to an integer with *n* digits, $e_{1j} \ldots e_{nj}$ in the base (x+1), where

$$e_{ij} = \begin{cases} 0, if\ v_i \notin \mathrm{S}_j \\ 1\ \ if\ v_i \in \mathrm{S}_j \end{cases}$$

- So, we have *x* integers, where

$$a_j = \sum_{r=1}^{n} e_{rj}\,(x+1)^{r-1}$$

Exact-cover $\leq_p$ Knapsack

- The sum of all the numbers assigned to an exact cover, *k* can be defined as

$$k = \sum_{r=1}^{n} (x+1)^{r-1}$$

- The instance of knapsack problem has a solution iff the initial instance of the exact cover has a solution

Exact-cover $\leq_p$ Knapsack

- The solution is polynomial time reducible because it only requires polynomial time $p(n)$ to translate the input of exact cover to knapsack

- Conversion of the subsets to its binary equivalent has a time complexity of $O(n^2)$

- Conversion of binary representation to base $(x+1)$ has a time complexity of $O(n)$

Exact-cover $\leq_p$ Knapsack

- Let *S = {A, B, C, D, E, F}* be a collection of subsets of a set *X = {1, 2, 3, 4, 5, 6, 7}* such that:
  - v_1 *= A = {1, 4, 7} = 1001001 = 117,993*
  - v_2 *= B = {1, 4} = 1001000 = 117,992*
  - v_3 *= C = {4, 5, 7} = 0001101 = 393*
  - v_4 *= D = {3, 5, 6} = 0010110 = 2,457*
  - v_5 *= E = {2, 3, 6, 7} = 0110011 = 19,216*
  - v_6 *= F = {2, 7} = 0100001 = 16,808*
- *K = {1, 2, 3, 4 ,5 ,6 ,7} = 1111111 = 137,257*
- $v_2 + v_4 + v_6 = K$; one of the solution of 0/1 Knapsack problem.

Thank You