

CS F364

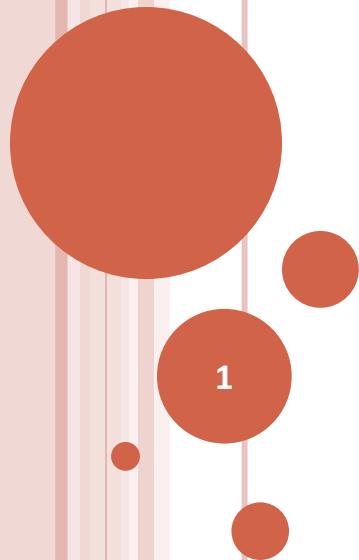
Design & Analysis of Algorithms

ALGORITHM DESIGN TECHNIQUES

Divide & Conquer:

Memoization

- Example: 0,1 Knapsack



MEMOIZATION

- Memo (store) the results of the sub-problems
 - with the hope that *they will be re-used*
- Space-Time tradeoff:
 - If the results are reused time is saved
 - If the results are not reused space is wasted
- When do we know that results will be reused?
 - Structure of the problem
 - If the problem can be specified in terms of the sub-problems
 - then we can identify whether sub-problems overlap.

PROBLEM – 0/1 KNAPSACK

○ Given:

- A sack with max. capacity by weight: W kg.
- Set S of items j (in store) labeled with
 - Weight : w_i ($< W$)
 - Price : p_j

○ Assumption:

- An item is either taken (in full) or not
- All values (w_j , p_j , and W) are positive.

○ Goal:

- Fill the sack with maximum value (by price)
 - i.e. Find T subset of S , such that
 - $\sum_{i \in T} p_i$ is maximum and $\sum_{i \in T} w_i \leq W$

KNAPSACK – DIVIDE-AND-CONQUER

- Algorithm KnapSack(S, W) // S - Set of items; W -capacity
 1. Find all the subsets of S say T_1, T_2, \dots, T_k
 1. Let the cumulative weights be $w(T_1), w(T_2), \dots, w(T_k)$ and prices be $p(T_1), p(T_2), \dots, p(T_k)$
 2. return (T_m, v) such that
$$v = \max \{ p(T_i) \mid w(T_i) \leq W \text{ and } 1 \leq i \leq k \}$$
and T_m such that $w(T_m) = v$

What is the time complexity of this algorithm?

MEMOIZED KNAPSACK

1. Let $|S|$ be n ; Denote items in S as j_1, j_2, \dots, j_n
2. Let the subsets of $\{j_1, j_2, \dots, j_{n-1}\}$ be T_1, T_2, \dots, T_k
 1. Let the cumulative weights be $w(T_1), w(T_2), \dots, w(T_k)$ and prices be $p(T_1), p(T_2), \dots, p(T_k)$
 2. Let $Sol_{n-1} = (T_m, v)$ such that
$$v = \max \{ p(T_i) \mid w(T_i) \leq W \text{ and } 1 \leq i \leq k \} = p(T_m)$$
3. Compute $P_n = \{ p(T_i) + p_n \mid w(T_i) + w_n \leq W \text{ and } 1 \leq i \leq k \}$
 - Compute (T_m', v') s.t. $v' = \max(P_n) = p(T_m') + p_n$
4. return the pair with maximum value among (T_m, v) and (T_m', v')

What is the time complexity of this algorithm?