Subset Sums and Knapsacks

Dynamic Programming - II

Subset Sums

- **Problem:** Given n items {1, 2, … , n}, and each has a given non-negative weight w_i (for i={1, 2, … , n}). We also have a bound W. We would like to select a subset S if the items so that $\sum_{i \in S} w_i \leq W$, and subject to the restriction that $\sum_{i \in S} w_i$ is as large as possible.
- Informally: Consider, we have a single machine that we can run from 0 to W time to process jobs. If we have n jobs to process where each job requires w_i time, to keep the machine as busy as possible, which jobs will you choose? => you want maximum resource utilization.

Knapsack Problem

- **Problem:** Given n items {1, 2, … , n}, and each has a given non-negative weight w_i and an associated value v_i (for i={1, 2, … , n}) . We also have a bound W. We would like to select a subset S if the items so that $\sum_{i \in S} w_i \leq W$, and subject to the restriction that $\sum_{i \in S} v_i$ is as large as possible.
- Informally: Consider, we have a knapsack that can hold upto W kgs. If we want to choose from n items to add to the bag where each item is w_i kgs and costs v_i INR, to keep the knapsack as valueable as possible, which items will you choose? => you want maximum resource utilization.
- **More general case of Subset Sum Problem**

Will greedy approach work?

Give counter examples to show that greedy will not for:

1) Sort the items by decreasing weight
2) Sort by increasing weight

+2 in Quiz1 for first 3 students to give correct answer (over mail or in class)