

Step 2: To show that Knapsack is NP-complete we shall reduce SubsetSum to Knapsack SubsetSum Instance: A sequence of positive integers $S = \{a_1, \dots, a_n\}$ and a target integer t. Question: Is there a subset $T \subseteq S$ such that $\sum_{n \in T} a_i = t$?

Given a SubsetSum instance, that is a set $S=\{a_1,a_2,\dots,a_n\}$ and a target number t• Set $v_i=w_i=a_i$ • Set t=l=tThen for any subset $T\subseteq S$ $\sum_{i\in T} a_i = t \quad \text{if and only if} \quad \sum_{i\in T} v_i = \sum_{i\in T} a_i \geq t \quad \text{and} \quad \sum_{i\in T} w_i = \sum_{i\in T} a_i \leq t$

Polynomial Algorithm

Knapsack can be solved in O(nl) time using dynamic programming

• Create a $(l+1)\times (n+1)$ matrix M• Set M(w,0) and M(0,i) to 0 for all $w\in\{1,2,...,l\}$ and $i\in\{1,2,...,n\}$ • For i=1,2,...,n, set entry M(w,i) as follows $M(w,i+1)=\max\{M(w,i),M(w-w_{i+1},i)+v_{i+1}\}$ (M(w,i) is the largest total value obtainable by selecting from the first i items with weight l imit w• Answer yes if $M(l,n+1) \geq t$, otherwise no

Computability and Complexity **Example Construction** Let $V=\{1,3,4,2\}, \quad W=\{1,1,3,2\}, \quad t=7 \quad \text{and} \ l=5$

Pseudo-polynomial Time

This algorithm is not sufficient to show that Knapsack is in P!

The length of the input to Knapsack is in O(nlog I)—so nI is not bounded by a polynomial function of the input length

• An algorithm which is polynomial in the size of the numbers in the input is called pseudo-polynomial

• NP-complete problem which remains NP-complete when all numbers in the input are bounded by some polynomial in the length of the input is called strongly NP-complete¹

*See Garey and Johnson for more discussion of strong NP-completeness

Computability and Complexity 17-8

Strongly NP-complete Problems

- Any NP-complete problem without numerical data is strongly NP-complete: SAT, HamiltonianCircuit, other graph problems
- TSP(D) is strongly NP-complete

(From the reduction of HamCircuit we know that even if the distances between cities are bounded by a linear polynomial, the problem remains equivalent to HamCircuit)

· SubsetSum?

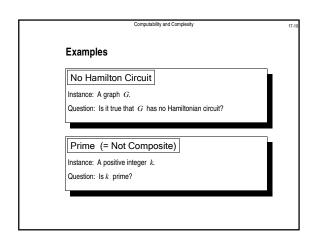
NO! It is a subproblem of Knapsack

NP and coNP

For a language L over an alphabet Σ, we denote the complement of L, the language Σ*– L

Definition
The class of languages L such that has a polynomial time verifiers is called coNP

In other words, a problem belongs to coNP if the no-instances have succinct certificates



More Examples

Validity

Instance: A conjunctive normal form Φ.
Question: Is Φ valid?

