CS341: Algorithms
Lecture 22: Intractability IV

Yao-Liang Yu
School of Computer Science
University of Waterloo

Mar 28, 2017
SUBSETSUM

Instance: A list of positive integers $S = [s_1, \ldots, s_n]$ and a target sum integer $T$.

Question: Does there exist a subset $J \subseteq \{1, \ldots, n\}$ such that $\sum_{j \in J} s_j = T$?

Theorem. SUBSETSUM $\in \text{NPC}$.

Proof. Clearly SUBSETSUM $\in \text{NP}$.

Next, we show VERTEXCOVER $\leq_P$ SUBSETSUM.

Take a graph $G$ with nodes $V = \{v_1, \ldots, v_n\}$ and edges $E = \{e_1, \ldots, e_m\}$, and an integer $1 \leq k \leq n$. Let

$$l_{ij} = \begin{cases} 1, & v_i \in e_j \\ 0, & \text{otherwise} \end{cases}$$
Proof cont’

<table>
<thead>
<tr>
<th></th>
<th>(e_1)</th>
<th>(e_2)</th>
<th>(\cdots)</th>
<th>(e_{m-1})</th>
<th>(e_m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\nu_1)</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>(\cdots)</td>
<td>0</td>
</tr>
<tr>
<td>(\nu_2)</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>(\cdots)</td>
<td>0</td>
</tr>
<tr>
<td>(\vdots)</td>
<td>(\vdots)</td>
<td>(\vdots)</td>
<td>(\vdots)</td>
<td>(\vdots)</td>
<td>(\vdots)</td>
</tr>
<tr>
<td>(\nu_n)</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>(\cdots)</td>
<td>1</td>
</tr>
<tr>
<td>(e_1)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>(\cdots)</td>
<td>0</td>
</tr>
<tr>
<td>(e_2)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>(\cdots)</td>
<td>1</td>
</tr>
<tr>
<td>(\vdots)</td>
<td>(\vdots)</td>
<td>(\vdots)</td>
<td>(\vdots)</td>
<td>(\vdots)</td>
<td>(\vdots)</td>
</tr>
<tr>
<td>(e_{m-1})</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>(\cdots)</td>
<td>0</td>
</tr>
<tr>
<td>(e_m)</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>(\cdots)</td>
<td>0</td>
</tr>
</tbody>
</table>

|       | \(k\) | 2      | 2         | \(\cdots\) | 2      | 2      |

Note. Can add redundant nodes to a vertex cover (to increase its size).

\[
T = k \cdot 10^m + \sum_{j=0}^{m-1} 2 \cdot 10^j.
\]
A dynamic programming solution

Let $B(i, t)$ be true if we can find a subset of the first $i$ numbers whose sum is $t$, and false otherwise.

Recursion: $B(i, t) = B(i - 1, t) \lor B(i - 1, t - a_i)$.

Base case: $B(1, t) = 1$ iff $t = a_1$.

The table $B$ has size $\Theta(nT)$ hence the running time $O(nT)$.

This is polytime iff $T = \text{poly}(n)$.

**Implication:** For any $\Pi \in \text{NPC}$, if $\Pi \leq_p \text{SUBSETSUM}$, then we can we say about $T$ in the reduction?

**Implication:** If $\text{SUBSETSUM} \leq_p \Pi$ but $T = \text{poly}(n)$ in the reduction, can we conclude $\Pi \in \text{NP-hard}$?
Component Grouping

**Component Grouping (CG)**

**Instance:** A disconnected graph $G$ and a positive integer $k$.

**Question:** Does there exist a subset of the connected components of $G$ whose union has size exactly $k$?

**Theorem.** $\text{CG} \in \text{NPC}$. 

**Proof.** Clearly $\text{CG} \in \text{NP}$. We show $\text{SUBSETSUM} \leq_P \text{CG}$.

Take an instance of $\text{SUBSETSUM}$. Construct disjoint paths $P_i$ that have length $s_i$, respectively. Easy to see $\text{SUBSETSUM}$ is true iff $\text{CG}$ is true (with $k = T$).

Something is very wrong here!!!

In fact, $\text{CG} \in \text{P}$. Why?

Y-L. Yu (UW)  
Intractability IV: Numeric  
Mar 28, 2017 5 / 8
Component Grouping

**Instance:** A disconnected graph $G$ and a positive integer $k$.

**Question:** Does there exist a subset of the connected components of $G$ whose union has size exactly $k$?

**Theorem.** $\text{CG} \in \text{NPC}$.

Proof. Clearly $\text{CG} \in \text{NP}$. We show $\text{SUBSETSUM} \leq_P \text{CG}$.

Take an instance of $\text{SUBSETSUM}$. Construct disjoint paths $P_i$ that have length $s_i$, respectively. Easy to see $\text{SUBSETSUM}$ is true iff $\text{CG}$ is true (with $k = T$).

**Something** is very wrong here!!!
Component Grouping

**Instance:** A disconnected graph $G$ and a positive integer $k$.

**Question:** Does there exist a subset of the connected components of $G$ whose union has size exactly $k$?

**Theorem.** $\text{CG} \in \text{NPC}$.

Proof. Clearly $\text{CG} \in \text{NP}$. We show $\text{SUBSETSUM} \leq_{P} \text{CG}$.

Take an instance of $\text{SUBSETSUM}$. Construct disjoint paths $P_i$ that have length $s_i$, respectively. Easy to see $\text{SUBSETSUM}$ is true iff $\text{CG}$ is true (with $k = T$).

**Something** is very wrong here!!!

In fact, $\text{CG} \in \text{P}$. Why?
Knapsack

**KNAPSACK**

**Input:** Profits $P = [p_1, \ldots, p_n]$; weights $W = [w_1, \ldots, w_n]$; capacity $C$.

**Feasible solution:** $X = [x_1, \ldots, x_n]$ such that $\sum_i w_i x_i \leq C$, for all $i$, $x_i \in \{0, 1\}$.

**Output:** A feasible solution $X$ that maximizes $\sum_i p_i x_i$.

Optimization to Decision.

**Theorem.** KNAPSACK $\in$ NPC.

Proof. Clearly KNAPSACK $\in$ NP.

We show SUBSETSUM $\leq_P$ KNAPSACK.

$P = W = S$, $C = T = k$. 
**Partition**

**PARTITION**

**Instance:** A list $Q = [q_1, \ldots, q_m]$ of positive integers.

**Question:** Can we partition $Q$ into two subsets whose sum equal?

**Theorem.** $\textsc{Partition} \in \text{NPC}$.  

Proof. Clearly $\textsc{Partition} \in \text{NP}$. We show $\textsc{SubsetSum} \leq_P \textsc{Partition}$.

Take an instance of $\textsc{SubsetSum}$ and let

$$Q = [s_1, \ldots, s_n, 2 \sum_i s_i - T, \sum_i s_i + T].$$

Key: If such partition exists, then $2 \sum_i s_i - T$ and $\sum_i s_i + T$ cannot be in the same subset.