Construct a Directed Acyclic Graph (DAG)

- Add an edge from $a[i] \rightarrow a[j]$ if $i < j$ and $a[i] < a[j]$.

Longest Increasing Sequence in $a[1] \ldots a[n] \iff$ Longest Path in DAG
**Longest Path in a DAG**

**Input:** DAG (assume 1, 2, 3, ..., n in the linearized order)

**Goal:** Find the longest path in DAG

**DP Algorithm:**

1) **Subproblem:**

\[ T[i] = \text{longest path ending at vertex } i. \]

2) **Reccurrence Relation:**

\[ T[i] = \max_{(j \rightarrow i)} \{ T[j] + 1 \} \]

3) **Compute** \( T[1], T[2], T[3], \ldots, T[n] \) (linearized order)
KNAP SACK (with REPETITION)

Input: List of objects with (weight, value) 

\[(w_1, v_1), (w_2, v_2), \ldots, (w_n, v_n)\]

Maximum Weight = \(W\).

Goal: Pick a subset of objects of total weight < \(W\) that maximizes total value.
KNAPSACK: (with repetition)

**Input:** A set of items \((w_1, v_1), (w_2, v_2), \ldots, (w_n, v_n)\)

**Goal:** Find a bundle of items with total weight \(W < W\) that has maximum value.

**Example:** \(W = 3\)

<table>
<thead>
<tr>
<th>Weight (lb)</th>
<th>Value ($)</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓ A</td>
<td>15</td>
</tr>
<tr>
<td>B</td>
<td>6</td>
</tr>
<tr>
<td>✓ C</td>
<td>7</td>
</tr>
<tr>
<td>✓ D</td>
<td>8</td>
</tr>
</tbody>
</table>

- \(A + C + 0\)
- \(15 + 7 + 8 \leq 30\)
- \(43 + 19 + 23 = 85\)
- \(A + A\)
- \(15 + 15 \leq 30\)
- \(93 + 23 = 86\)
Knapack (with repetition) $\rightarrow$ longest path in a DAG

Graph: Vertices $\{0, 1, 2, 3\}$, Weighted Edges: For each $w \in \{1, \ldots, W\}$, $w \in \mathbb{R}$

Claim: Largest Value Bundle $\rightarrow$ longest path in DAG

Total weight
**KNAP SACK:**

**Input:** A set of items $(w_1, v_1), (w_2, v_2), \ldots, (w_n, v_n)$.

- **Weight:** $w_i$  
- **Height:** $h_i$  
- **Value:** $v_i$  

**Goal:** Find a bundle of items with total weight $\leq W$ that has maximum value.

**Maximum Total Weight =** $W$  
**Maximum Total Height =** $H$
- **STEP 1:** Subproblem \(\{1,\ldots, n\}\)

\[ K[\omega, i] = \text{largest value bundle that has weight } \leq \omega \text{ and only items } \{1, \ldots, i\} \]

**RETURN:** \(K(W, n)\)

- **STEP 2:** Recurrence Relation

\[
K[\omega, h, i] = \max\left\{
\begin{array}{l}
K[\omega, i-1] \\
K(\omega - w_i, h - h_i, i-1) + v_i
\end{array}
\right\}
\]

**Case 1**
- \(i\) not picked

**Case 2**
- \(i\) picked
Order increasing weight \& items

\[\text{for } i = 1 \text{ to } n \rightarrow\]
\[\text{for weight } \omega = 1 \text{ to } W \rightarrow\]
\[K[\omega,i] = \max \begin{cases} K[\omega,i-1], & \text{if } \omega > \omega_i \\ K[\omega-\omega_i,i-1] + v_i \end{cases}\]

\[O(nW)\]

\[\text{pseudo polynomial}\]

\[K[\omega,i] = -\infty\]

\[\text{Base cases: if } \omega = 0 \text{ return } 0\]

\[\text{Return } \max \{ K[\omega,i-1], K[\omega-\omega_i,i-1] + v_i \}\]
\[ W = 2000 \]

<table>
<thead>
<tr>
<th>A</th>
<th>2000</th>
<th>2000</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>1001</td>
<td>1003</td>
</tr>
<tr>
<td>C</td>
<td>1001</td>
<td>1003</td>
</tr>
</tbody>
</table>

Subproblem:

\[ K[w, h, i] = \text{largest value bundle that uses total weight} \leq w \]

\[ \text{total height} \leq h \]

\[ \text{only} \] and item \( i \)