1. Correctness Proof for Knapsack

Proof by induction.

Say \((i, j) < (i', j')\) if \(i < i'\) or \((i = i' \text{ and } j < j')\)

\((0, 0) < (0, 1) < (0, 2) < \ldots < (1, 0) < (1, 1) < \ldots\)

IH: Induction hypothesis: \(a[i][j]\) is correct for all values of \(a[i'][j']\) where \((i, j) < (i', j')\)

Base case: \(a[0][0] = a[0][j] = 0\) for all \(i, j\)

Induction step: when comparing \(a[i'][j']\), by IH

\[a[i'-1][j'], a[i'-1][j'-w_i]\] are already computed correctly

\(a[g]\) considers the optimal value for item \(i'\) in knapsack

\[a[i'-1][j'-w_i] + v_i\]

for item \(i'\) not in knapsack

\[a[i'-1][j']\]

\(\implies\) Value at \(a[i'][j']\) is also correct.

- Longest Common Subsequence (LCS)

\[LCS(a[n], b[m])\]

- Last decision: whether \(a[n]\) should be in the LCS

\['ababcde\', 'labbe\cd']

4 possible cases

\(a[n]\) \[
\begin{array}{c}
\text{in} \\
\text{not in}
\end{array}
\]

Let \(LCS[i][j]\) be the length of LCS of \(a[1..i], b[1..j]\)
\[ a \in \{ \text{in}, \text{not in} \} \]

\[ c[n,m] = \begin{cases} 
  c[n-1,m] & \text{case 1 } a \in \text{not in LCS} \\
  c[n,m-1] & \text{case 2 } b[m] \text{ not in LCS} \\
  c[n-1,m-1] + 1 & \text{case 3 } \text{if } a[n] = b[m] \text{ both in LCS} 
\end{cases} \]

\[ \text{base case: if } i=0, j=0 \quad c[i,j]=0 \]

\[ \text{ordering: } i = 1 \to n \]
\[ j = 1 \to m \]

---

- **Maximum Independent Set on Trees**

  - **Goal:** Relate solution of the whole tree to solutions of the subtrees.

  - For the root \{ not in the independent set \text{ (take max indep. set for all children's subtrees) } \\
  \text{in the indep. set.} \}

  \[ F(u) = \max \text{ ind. set of subtree rooted at } u \]

  \[ C(u) = \text{ but } u \text{ cannot be in the set} \]
\[ F(u) = \max \left\{ \sum_{v \in \text{child of } u} F(v) \quad \text{(if } u \text{ is not in set)} \right\}
\]
\[ \quad \sum_{v \in \text{child of } u} G(v) + 1 \quad \text{(if } u \text{ is in the set)} \]

\[ G(u) = \sum_{v \in \text{child of } u} F(v) \quad \text{(same as case 1 for } F) \]