1 Rod cutting

- The problem: Given a rod of length n and a table of prices $p[i]$ for $i = 1, 2, 3, ..., n$, determine the maximal revenue obtainable by cutting up the rod and selling the pieces.
- Notation and choice of subproblem: We denote by $maxrev(x)$ the maximal revenue obtainable by cutting up a rod of length x. To solve our problem we call $maxrev(n)$.
- Recursive definition of $maxrev(n)$:

  maxrev(x)

  if $(x \leq 0)$: return 0

  For i = 1 to n: compute $p[i] + \texttt{maxrev}(x - i)$ and keep track of max

  RETURN this max

- Correctness: see notes.
- Dynamic programming solution, top-down with memoization:

  We create a table of size $[0..n]$, where $table[i]$ will store the result of $maxrev(i)$. We initialize all entries in the table as 0. To solve the problem, we call $maxrevDP(n)$.

  maxrevDP(x)

  if $(x \leq 0)$: return 0

  IF $table[x] \neq 0$: RETURN $table[x]$

  For i = 1 to n: compute $p[i] + \texttt{maxrevDP}(x - i)$ and keep track of max

  $table[x]$ = max

  RETURN $table[x]$

- Dynamic programming, bottom-up:

  maxrevDP_iterative(x)

  create $table[0..n]$ and initialize $table[i] = 0$ for all i

  for $(k = 1; k \leq n; k++)$

  for $(i = 1; i \leq k; i++)$

  set $table[k] = \texttt{max}\{table[k], p[i] + table[k - i]\}$

  RETURN $table[n]$

- Analysis: $O(n^2)$
- Computing full solution: