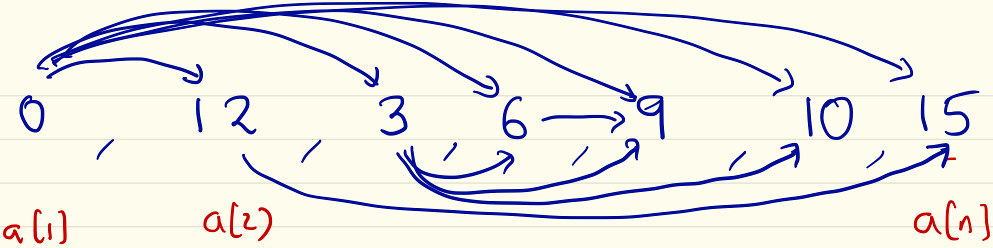
LECTURE 11



Construct a Directed Acyclic Graph (DAG)

"Add an edge from a[i] → a[j]

if $i < j$

and $a[i] < a[j]$ "

Longest Increasing Sequence in a[1] .. a[n] ⟺ Longest Path in DAG

LONGEST PATH IN A DAG

INPUT: DAG (assume $1 \to 2 \to 3 \ldots n$ is the linearised order)

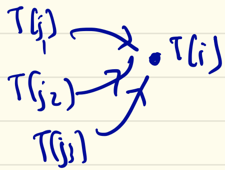
GOAL: Find the longest path in DAG

DP Algorithm: 1) Subproblem:

$T[i]$ = longest path ending at vertex i.

2) Reccurence Relation:

$$T[i] = \underset{(j \to i)}{\text{maximum}} \left\{ T[j] + 1 \right.$$



3) Compute $T[1], T[2], T[3] \ldots T[n]$ (linearised order)

KNAPSACK (with REPETITION)

Input: List of objects with (weight, value)

$(w_1, v_1), (w_2, v_2) \ldots (w_n, v_n)$

MaximumWeight $= W$.

Goal: Pick a subset of objects of total weight $< W$ that maximizes total value.

KNAPSACK: (with repetition)

INPUT: A set of items $(\underset{\text{weight}}{w_1}, \underset{\text{value}}{v_1})\ (w_2, v_2) \dots (w_n, v_n)$

Maximum Total Weight $= W$.

GOAL: Find a bundle of items with total weight $< W$ that has maximum value.

EXAMPLE: $W = 30$

	Weight (lbs)	Value ($)
✓ A	15	43
B	6	18
✓ C	7	19
✓ D	8	23

$A + C + D$

$15 + 7 + 8 \leq 30$

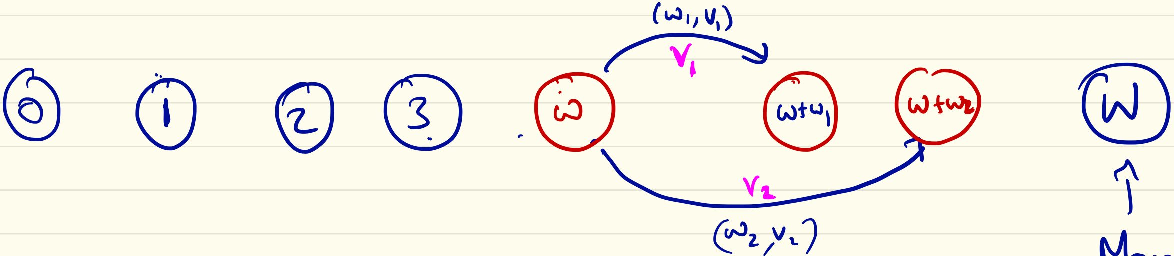
$43 + 19 + 23 = 85$

$A + A$

$15 + 15 \leq 30$

$43 + 43 = 86$

Knapsack (with repetition) $\longrightarrow$ longest path in a DAG



Maximum total weight

Graph : Vertices $\{0,1 \ldots W\}$ ← (all integers less than W)

Weighted Edges: For each $w \in \{1 .. W\}$

For each $i \in \{1 .. n\}$

Add $w \xrightarrow{v_i} w+w_i$ } Wn time

Claim: Largest Value Bundle

= $\underbrace{\text{longest}}_{\text{Total weight}}$ path in DAG

KNAPSACK: (NO repetition)

INPUT: A set of items $(\underset{h_1}{\overset{\text{weight}}{w_1}}, v_1)\ (w_2, v_2)_{h_2} \ldots (w_n, v_n)_{h_n}$

height: h_1 value

Maximum Total Weight $= W$.

Maximum Total Height $= H$

GOAL: Find a bundle of items with total weight $< W$ total Height $\leq H$ that has maximum value.

- STEP 1: Subproblem

$\{1..n\}$

$K[w, i] =$ largest value bundle that has weight $\leq w$ and uses only items $\{1..i\}$

$\{1, .. W\}$

RETURN: $K(W, n)$

- STEP 2: Reccurrence Relation

$K[w,i] = \max \begin{cases} \text{Case 1: item } i \text{ not picked} & K(w, i-1) \quad \{1.. i-1\} \\ \text{Case 2: item } i \text{ is picked} & K(w - w_i, i-1) + v_i \quad \{1.. i-1\} \end{cases}$

$K[w, h, i]$

$K(w, h, i-1)$

$K(w - w_i, h - h_i, i-1) + v_i$

-

$K[\omega, i]$

- Order increasing weight & items

for $i = 1$ to n

for weight $\omega = 1$ to W

$$K[\omega, i] = \max \begin{cases} K[\omega, i-1], & 0 \\ \text{if } (\omega > \omega_i) \\ K[\omega - \omega_i, i-1] + v_i & 1 \end{cases}$$

2

$O(nW)$

pseudopolynomial

$K[\text{-ve number}, \ ] = -\infty$

$K[\omega, i]$ { Base cases: if $\omega = 0$ return 0

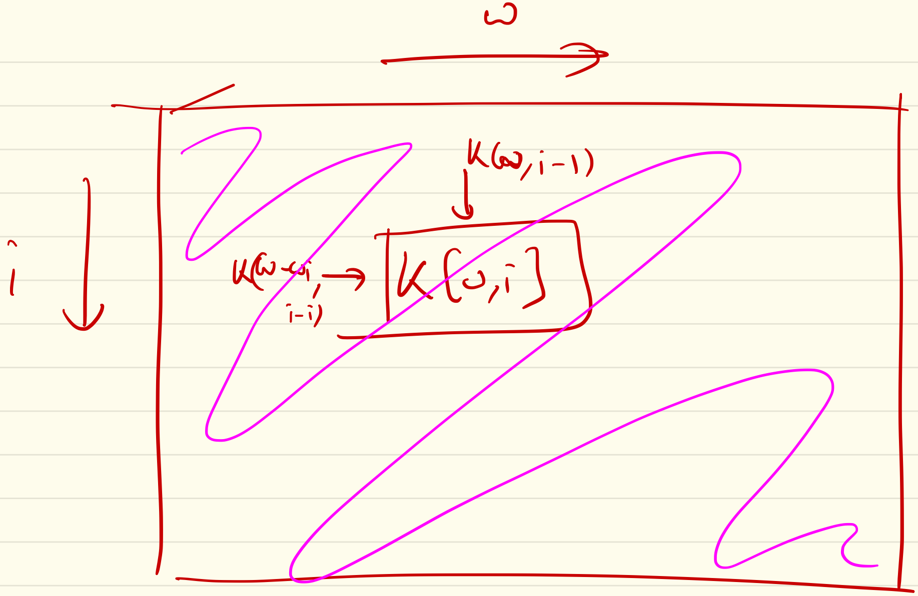
Return $\max\{K(\omega, i-1), K(\omega - \omega_i, i-1) + v_i\}$ }

$$\boxed{W = 2000}$$

A	2000	2000
B	1000	1002
$\rightarrow$ C	1001	1003

Subproblem:

$K[w, h, i]$ = largest value bundle that uses
total weight $\leq w$
total height $\leq h$
and only items $\{1 .. i\}$



ω
i
K(ω, i−1)
K(ω−ω_i, i−1)
K[ω, i]