Review - 2

Quiz
Syllabus:
Dynamic Programming

0-1 Knapsack

→ Items $1, 2, \ldots, n$
Weights $w_1, \ldots, w_n$ $\in \mathbb{Z}$
Value $v_1, \ldots, v_n$

Constraint: Knapsack has a capacity of W. $\in \mathbb{Z}$

maximize the value of items in the sack.

$$\max_{S \subseteq [n]} \sum_{i \in S} v_i$$

subj: $\sum_{i \in S} w_i \le W$

Say item 1 (n) is chosen
$(2, \ldots, n)$

$$\max_{S' \subseteq [2,n]\ ([1,n-1])} \sum_{i \in S'} v_i$$

subj: $\sum_{i \in S'} w_i \le W - w_1$ (w_n)

If item 1 (n) is not chosen

$$\max_{S'' \subseteq [2,n]\ ([1,n-1])} \sum_{i \in S''} v_i$$

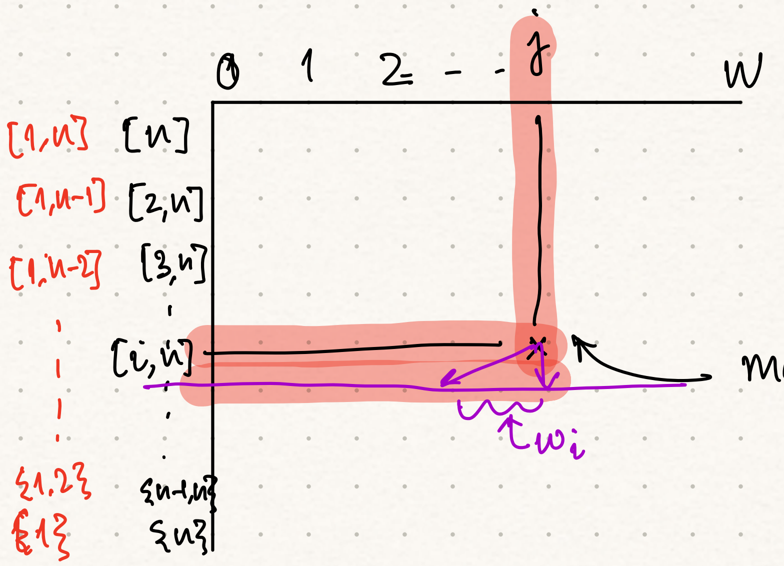
subj: $\sum_{i \in S''} w_i \le W$

$$\mathrm{Opt}([1,n], W) = \max \begin{cases} v_1\ (v_n) + \mathrm{OPT}([2,n]\ ([1,n-1]), W - w_1\ (w_n)) \\ \mathrm{OPT}([2,n]\ ([1,n-1]), W) \end{cases}$$

Sub Problems: $\mathrm{OPT}([*, n], *)$

# of choices is n

no. of choices is $\underline{W+1}$
$[0, W]$



$$\mathrm{OPT}([i,n], j)$$

$$\max \begin{cases} \mathrm{OPT}([i+1,n], j - w_i) + v_i \\ \mathrm{OPT}([i+1,n], j) \end{cases}$$

Last row is indexed by $\{n\}$.

$$OPT(\{n\}, j) = \begin{cases} v_n & \text{if } w_n \le j, \\ 0 & \text{o/w} \end{cases}$$

$$OPT(\{n-1, n\}, j) = \max \begin{cases} OPT(\{n\}, j) \\ OPT(\{n\}, j - w_{n-1}) + v_{n-1} \end{cases}$$

Computation goes column by column, from left to right.
and in the column entry by entry (order given by dependencies)

$\boxed{[n] \setminus S}$

Min Cost Triangulation

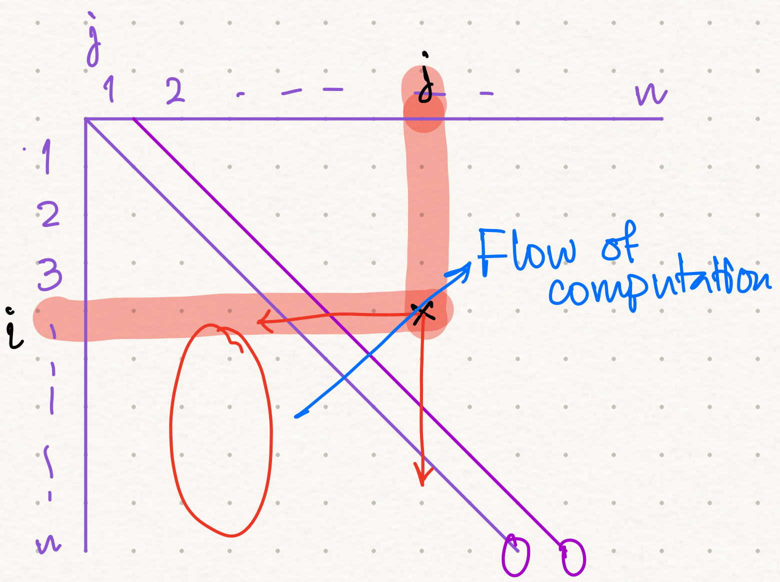
$[1, k]$ $[k, n]$

$[1,$

$k, k+1, \ldots, j-1, j$

$$\hookrightarrow \quad Cost([i, j]) = \min_{i<k<j} \{ cost([i, k]) + Cost([k, j]) + (w_{ik} + w_{kj} + w_{ij})$$

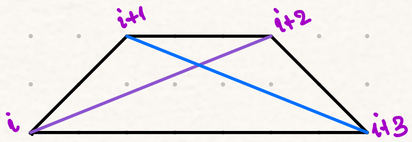
$i, i+1, i+2, \ldots, k-1, k$



Fill the matrix in the order of the values of $j-i$.

For $j-i = \{0, 1\}$ all values are zero.

$j - i = 3$



$$i, \underbrace{i+1, i+2}_{k}, i+3 = j$$

$$\text{Cost}[i,j] = \min_{k} \{ \text{cost}[i,k] + \text{cost}[k,j] + P_\Delta(i,k,j) \}$$

$$= \min \begin{cases} \cancel{\text{cost}[i,i+1]}^{\,0} + \text{cost}[i+1, i+3] + P_\Delta(i, i+1, i+3) \\ \text{cost}[i,i+2] + \cancel{\text{cost}[i+2,i+3]}_{\,0} + P_\Delta(i, i+2, i+3) \end{cases}$$

All Pairs Shortest Paths

weighted graph
(no negative cycles)

Want: Shortest paths between any pair of vertices.

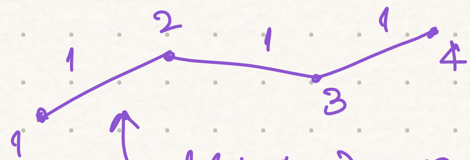
↳ n – instantiations of Djikstra's.

↳ $|V| \cdot (|E| \log |V|)$

↳ Worst case $\sim n^3 \log n$

$d(i,j,\ell)$: Shortest distance achieved over all i to j paths of length at most ℓ.

$$d(i,j,\ell) = \min_{k:(k,j)\in E} \begin{cases} d(i,j,\ell-1) \\ d(i,k,\ell-1) + w_{kj} \\ w_{i,j} \end{cases}$$



$d(1,4,2) = \infty$

$d(1,3,1) = \infty$

$d(1,4,3) = 3$.

For ℓ in $[1,n]$: } n iterations

  For i in $[1,n]$: } n iterations

n iterations → For j in $[1,n]$:

    $A = d(i,j,\ell-1)$

≤ max degree iterations ← For k in V s.t. $(k,j) \in E$:

$\sum_{j=1}^{n}$ # incoming neighbours

$\leq \underline{2m}$ iterations

iterations

$B = d(i,k,\ell-1) + w_{k,j}$

$C = w_{i,j}$

$d(i,j,\ell) = \min(A, B, C)$

$O(n^2 \cdot m)$

$\geq (nm\log n)$ given by n djikstra's

Optimizing this:

$$d(i,j,\ell) = \min_{k \in V} \left\{ d(i,k,\tfrac{\ell}{2}) + d(k,j,\tfrac{\ell}{2}) \right\}$$

For ℓ in $[1, \log n]$: ← $\log n$

For i in $[1,n]$: V ← n

For j in $[1,n]$: V ← n

MinValue = ∞

for k in V: ← n

$A = d(i,k,\tfrac{\ell}{2}) + d(k,j,\tfrac{\ell}{2})$

If MinValue > A:

MinValue ← A

$d(i,j,\ell)$ = MinValue.

$O(n^3 \log n)$

$\hat{d}(i,j,k)$: Shortest distance over all paths between i and j which only use vertices $\{1,2,\cdots,k\}$ as intermediate nodes.

$$\hat{d}(i,j,k) = \min \begin{cases} \hat{d}(i,j,k-1) \\ \hat{d}(i,k,k-1) + \hat{d}(k,j,k-1) \end{cases}$$

$\hat{d}(i,j,n)$ ← For all i,j we want this

If Shortest path uses vertex n

$\hat{d}(i,n,n-1) + \hat{d}(n,j,n-1)$

min

If shortest path does not use vertex n

$\hat{d}(i,j,n-1)$.

$\hat{d}(i,j,k)$

Fill memoization array in increasing order of k.