NOTE BOOK

Rokshana Ahmed

ALG(I) :

- Let d be the depth(I) n^2
- Sort the $|I| = n$ intervals increasingly by their starting time. $O(n \log n)$
- Let $I = \{(s_1, f_1), (s_2, f_2), \dots, (s_n, f_n)\}$ with $s_1 \le s_2 \le s_3 \le \dots \le s_n$
- For $j = 1$ to n
  - $L \leftarrow \{1, 2, \dots, d\}$
  - For $i = 1$ to $j-1$

    if (s_i, f_i) is incompatible with (s_j, f_j):

    $L \leftarrow L - \{\ell(i)\}$
  - IF $|L| \geq 1$:

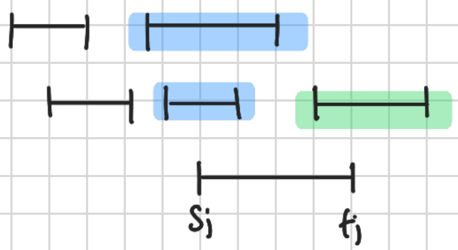
    Let $a \in L$
    Set $\ell(j) = a$
  - ELSE

    FAIL
- RETURN THE LABELING $\ell(1), \ell(2), \dots, \ell(n)$

L2 : The algorithm never fails.

P : Consider the generic iteration of the outer loop.
Let j be the value of that loop's index j in this iteration.
Let S_j be the set of intervals that the algorithm considered before the jth interval(s_j, f_j) and that (II) end after f_j.

$$S_j = \{(s_i, f_i) \mid i \le j-1 \text{ and } f_i \geq s_j\}$$



S_j is the set of intervals that we have already labelled and that interfere with (s_j, f_j).

Each interval in S_j will have a label that the inner loop will remove from L. No other label will be removed from L.

Thus, after the inner loop,

$$|L| \geq d - |S_j|,$$

since L started with d labels.

CLAIM: $|S_j| \leq d-1$

P: Each interval in S_j passes through the s_j, and also comes before (s_j, f_j) in the ordering.
Then $(s_j, f_j) \notin S_j$.
Suppose, by contradiction, that $|S_j| \geq d$.
Then, since $(s_j, f_j) \in S_j$, the set

$$T = S_j \cup \{(s_j, f_j)\} \text{ has cardinality } |T| \geq d+1$$

But, each interval in T passes through the s_j.

Then, $\text{DEPTH}(I) \geq |T| \geq d+1$. This contradicts $d = \text{DEPTH}(I)$ ■

But, then, after the loop, $|L| > d - |S_j| \geq 1$. Thus, $L \neq \emptyset$ and the algorithm doesn't fail ■

L3: The algorithm returns a valid labeling.

P: Suppose that (s_i, f_i) and (s_j, f_j) are overlapping intervals, with $i < j$.
Then, the label of (s_i, f_i) is chosen before the label of (s_j, f_j).
Since (s_i, f_i) and (s_j, f_j) are overlapping, the label we assign to (s_j, f_j) cannot be the label of (s_i, f_i) – indeed we removed that label from L, before picking a label for (s_j, f_j) from L.

T: The algorithm returns an optimal solution.

(or valid)

P: No feasible solution can have fewer than DEPTH(I) resources (L1)
The solution returned by the algorithm uses DEPTH(I) resources (L2), and it is FEASIBLE / VALID (L3) ■

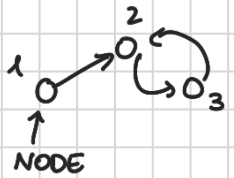
ALG(I) :

- Let d be the depth(I)
- Sort the $|I| = n$ intervals increasingly by their starting time.
- Let $I = \{(s_1, f_1), (s_2, f_2), \dots, (s_n, f_n)\}$ with $s_1 \leq s_2 \leq s_3 \leq \dots \leq s_n$
- For $j = 1$ to n
  - $L \leftarrow \{1, 2, \dots, d\}$
  - For $i = 1$ to $j-1$
    - IF (s_i, f_i) is incompatible with (s_j, f_j) : // if they overlap
      - $L \leftarrow L - \{\ell(i)\}$
  - // INVARIANT : $|L| \geq 1$ — comment to remind $|L| \geq 1$
  - Let $a \in L$
  - set $\ell(j) = a$
- Return the labeling $\ell(1), \ell(2), \dots, \ell(n)$

---

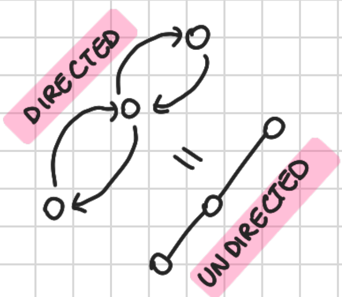
v: starting point
w: finish point

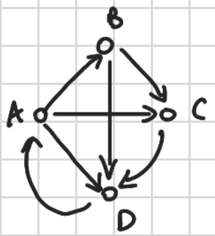
A (directed) graph $G(V, E)$ is composed of a set of vertices V and of a set of arcs E, with $E \subseteq \{(v, w) \mid v \neq w \text{ and } v, w \in V\}$



$V = \{1, 2, 3\}$

$E = \{(1,2), (2,3), (3,2)\}$

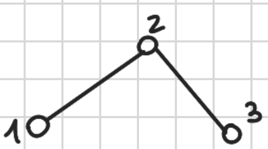




$V = \{A, B, C, D\}$

$E = \{(A,B), (B,C), (C,D), (D,A), (A,D), (A,C), (B,D)\}$

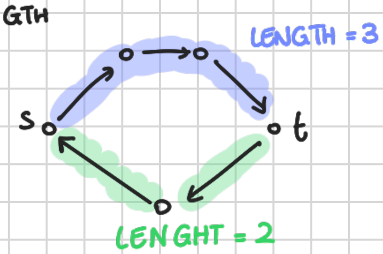
An undirected graph $G(V, E)$ is composed of a set of vertices V and of a set of edges E, with $E \subseteq \{\underline{(v, w)} \mid v \neq w \text{ and } v, w \in V\}$

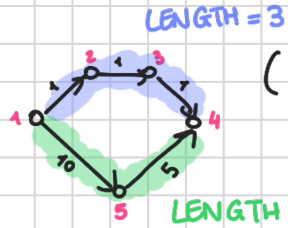


$V = \{1, 2, 3\}$

$E = \{(1,2), (2,3)\}$

A weighted directed graph is a directed graph $G(V,E)$ with a function $\ell : E \to \mathbb{R}_{\geq 0}$
LENGTH





(10 + 5 TOO MUCH PATH
1 + 1 + 1 QUICKER)

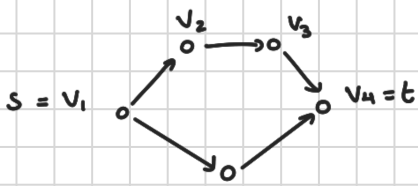
$V = \{1,2,3,4,5\}$

$E = \{(1,2), (2,3), (3,4), (1,5), (5,4)\}$

$$\left.\begin{array}{l} \ell(1,2) = 1 \\ \ell(2,3) = 1 \\ \ell(3,4) = 1 \\ \ell(1,5) = 10 \\ \ell(5,4) = 5 \end{array}\right\} \text{LENGTH OF PATHS}$$

HOW TO FIND THE SHORTEST PATH FROM A NODE "s" TO A NODE "t" IN THE GRAPH?

DEF: Given $G(V,E)$ a path from $s \in V$ to $t \in V$ is a sequence of nodes $s = v_1, v_2, v_3, \dots, v_k = t$ such that $(v_1, v_2) \in E$, $(v_2, v_3) \in E$, $\dots$, $(v_{k-1}, v_k) \in E$.

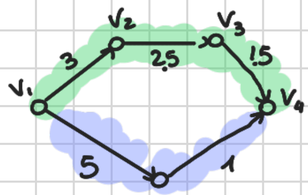


$\pi = v_1, v_2, v_3, v_4$ is a path from $v_1 = s$ to $v_4 = t$

$v_1 v_3 v_4$ is not a path
NEIN

DEF: If $G(V,E)$, ℓ is a weighted graph and if $\pi = v_1, v_2, \dots, v_k$ is a path in $G(V,E)$, then the length of π is:

$$\ell(\pi) = \sum_{i=1}^{k-1} \ell(v_i, v_{i+1})$$



$\pi = v_1, v_2, v_3, v_4$
$\ell(\pi) = 3 + 2{,}5 + 1{,}5 = 7$

$\pi = v_1, v_5, v_4$
$\ell(\pi) = 5 + 1 = 6$

PROBLEM: Given a weighted graph $G(V,S), \ell$, and given $s,t \in V$, what is the length of the shortest path from s to t?

E.W. DIJKSTRA'S ALGO ($G(V,E), \ell, s$) # figures out the shortest path

// the algorithm will use the set S to denote the
// set of nodes it visited so far.
// Moreover, $\forall u \in V$, $d(u)$ will be set to the length of a
// shortest path from "s" to "u".

$S \leftarrow \{s\}$

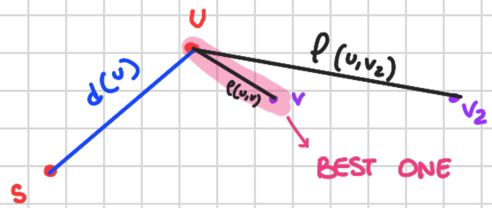
$d(s) = 0$ ← distance

WHILE $S \neq V$: # visiting all vertices

- SELECT A NODE $v \in V - S$ (a node yet not visited) THAT CAN BE REACHED DIRECTLY FROM SOME NODE IN S, AND FOR WHICH

$$d'(v) = \min_{\substack{(u,v) \in E \\ u \in S}} (d(u) + \ell(u,v)) \text{ is MINIMUM}$$

- $S \leftarrow S \cup \{v\}$
- $d(v) = d'(v)$



RETURN d

BECAUSE

$d(u) + \ell(u,v) < d(u) + \ell(u,v_2)$

↑ THIS IS THE MINIMUM