10/11/18 DIJKSTRA'S ALGORITHM:

(Doubly) Linked List Q

$O(n)$ Initialize (G, s) → for each vertex v: $v.d := \infty$, $v.\pi = \text{nil}$; $s.d := 0$

$O(n)$ $Q :=$ all vertices

while $Q \neq$ empty:

for every node v: $O(n)$ $v := \text{EXTRACT\_MIN}(Q)$

for each edge (v,u) out of v:

m_v times $O(1)$ RELAX(v,u) → if $u.d > v.d + w(v,$ then $u.d := v.d + w($; $u.\pi := v$

m_v: # of outgoing edges from v

Overall Runtime with (doubly) linked list:

$$O(n) + \underbrace{\sum_{\text{node } v} (O(n) + m_v \cdot O(1))}$$

$$\underbrace{\sum_{\text{node } v} O(n) + \sum_{\text{node } v} m_v \cdot O(1)}$$

$$O(n^2) + O(m)$$

→ because $\sum_{\text{node } v} m_v = m$

$\boxed{\text{Overall: } O(n^2 + m)}$

★ Big-Oh notation so upper bound, we are <u>not</u> showin it to be $\Theta(n^2 + m)$

- Looking at an implementation using heaps.

	Heap	
$\Theta(n)$ ←	$O(n)$	INITIALIZE(G, S)
$\Theta(n)$ ←	$O(n)$	$Q := $ all vertices
for every node v	$O(\log n)$	while $Q \neq$ empty:
		$v :=$ EXTRACT_MIN(Q)
		for each edge(v,u) out of v:
	m_v times $O(\log n)$	RELAX(v,u)

Overall Complexity,

$$\underbrace{O(n) + \sum_{\text{node } v} \left(O(\log n) + m_v \cdot O(\log n) \right)}_{O(n\log n) + O(m\log n)}$$

→ Because $\sum_{\text{node } v} m_v = m$

→ $\boxed{\text{Heap Overall: } O(n+m)\log n}$

$\boxed{\text{Linked List Overall: } O(n^2+m)}$

In some places you may find $O(n+m)\log n$ as $O(m\log n)$. This is because it is a natural assumption that $m > n$

Two cases:

		Heap	Linked list
Sparse G:	$m \approx n$	$\boxed{O(n \log n)}$	$O(n^2)$
Dense G:	$m \approx n^2$	$O(n^2 \log n)$	$\boxed{O(n^2)}$

→ Proof of Correctness:

Claim - All nodes removed from Q have $v.d = \delta(s$

actual length of shortest path from s

Proof - Induction on k^{th} node removed from Q

Base case: $k = 1$ $\quad s.d = 0 = d(s,s)$

Ind. hyp: The first k nodes removed from Q have $v.d = d(s,v)$

want to show that the $(k+1)^{st}$ node removed from Q has $v.d = d(s,v)$

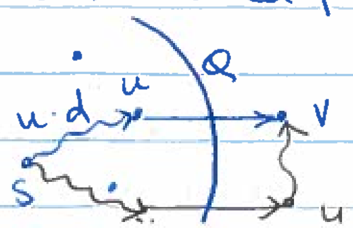
Let u be the node such that v.d was last changed when relaxing on (u,v)

$$v.d = u.d + w(u,v)$$

$$v.\pi = u$$

$u \notin Q$ since relax on edges out of u only called after u removed from Q

By ind. hyp, $u.d = d(s,u)$



Toward a contradiction, suppose ∃ path (black path) from s to v that's shorter than v.d

when x was removed from Q, we relaxed on (x, y).

So $y.d \leq x.d + w(x,y)$

By our ind. hyp. is $\delta(s,x)$

Since we chose v as the next node to remove from Q and not y, that means

$$\underbrace{v.d}_{= \delta(s,u) + w(u,v)} \leq y.d = \delta(s,x) + w(x,y)$$

length of blue path

1st part of black path upto y

This is a contradiction!

→ In the Interval problem,

we said out greedy algorithm returns

$i_1, i_2, i_3, \ldots, i_k$

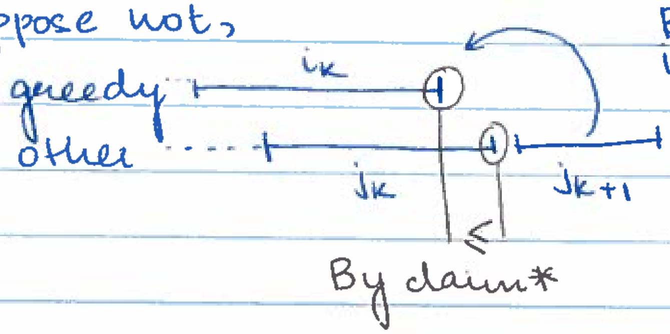
Any other set of mutually compatible courses: $j_1, j_2, j_3, \ldots, j_m$

stays ahead claim.

Claim*: $\forall r \leq k,\ f(i_r) \leq f(j_r)$

Claim: $m \leq k$

Proof: Suppose not,



But this j_{k+1} is compatible w/ i_k & greedy algorithm would have taken it