Dijkstra's algorithm (for single-source shortest paths problem)

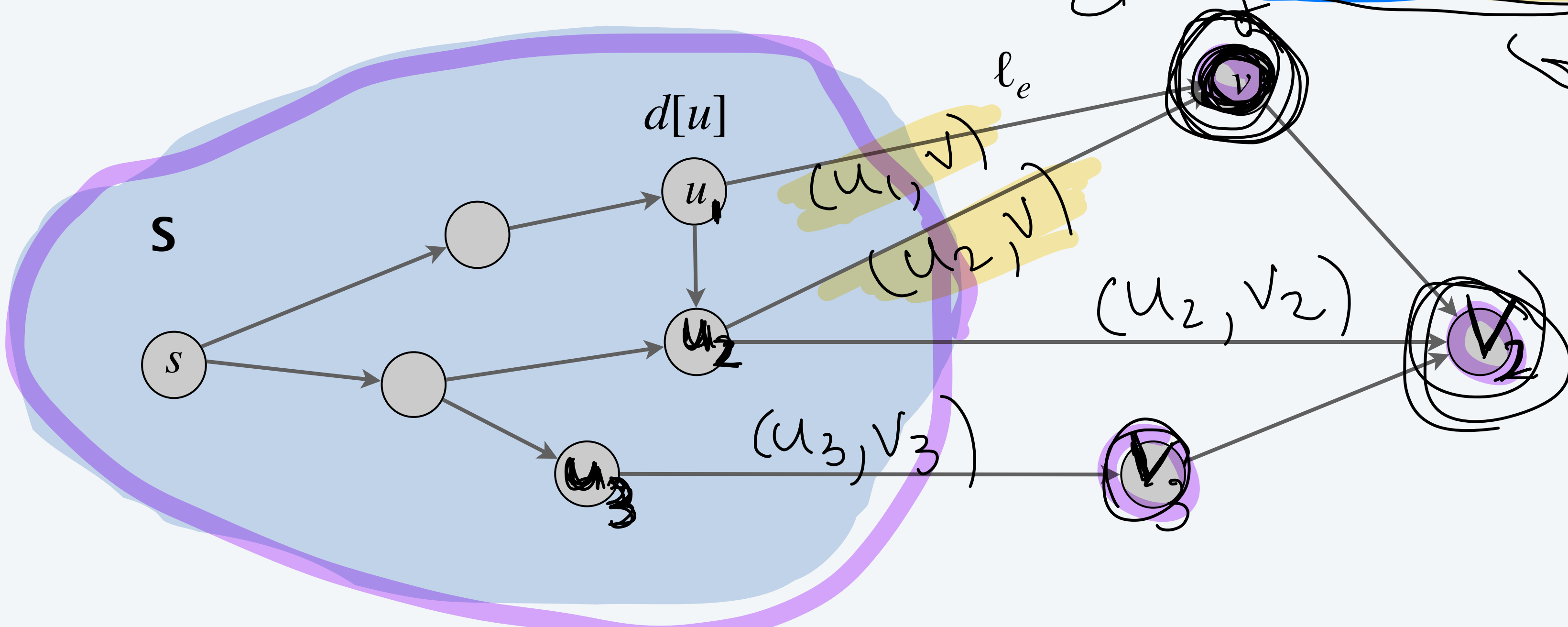
Greedy algorithm

Maintain a set of explored nodes S for which the algorithm has determined $d[u]$ = length of shortest $s \to u$ path.

Step 1: Initialize $S = \{s\}$, $d[s] = 0$.

Step 2: Repeatedly choose unexplored node $v \notin S$ which minimizes $d'(v) = \min_{e=(u,v):\,u\in S} (d(u) + \ell_e)$

minimum over all edges (u,v) such that u is in S



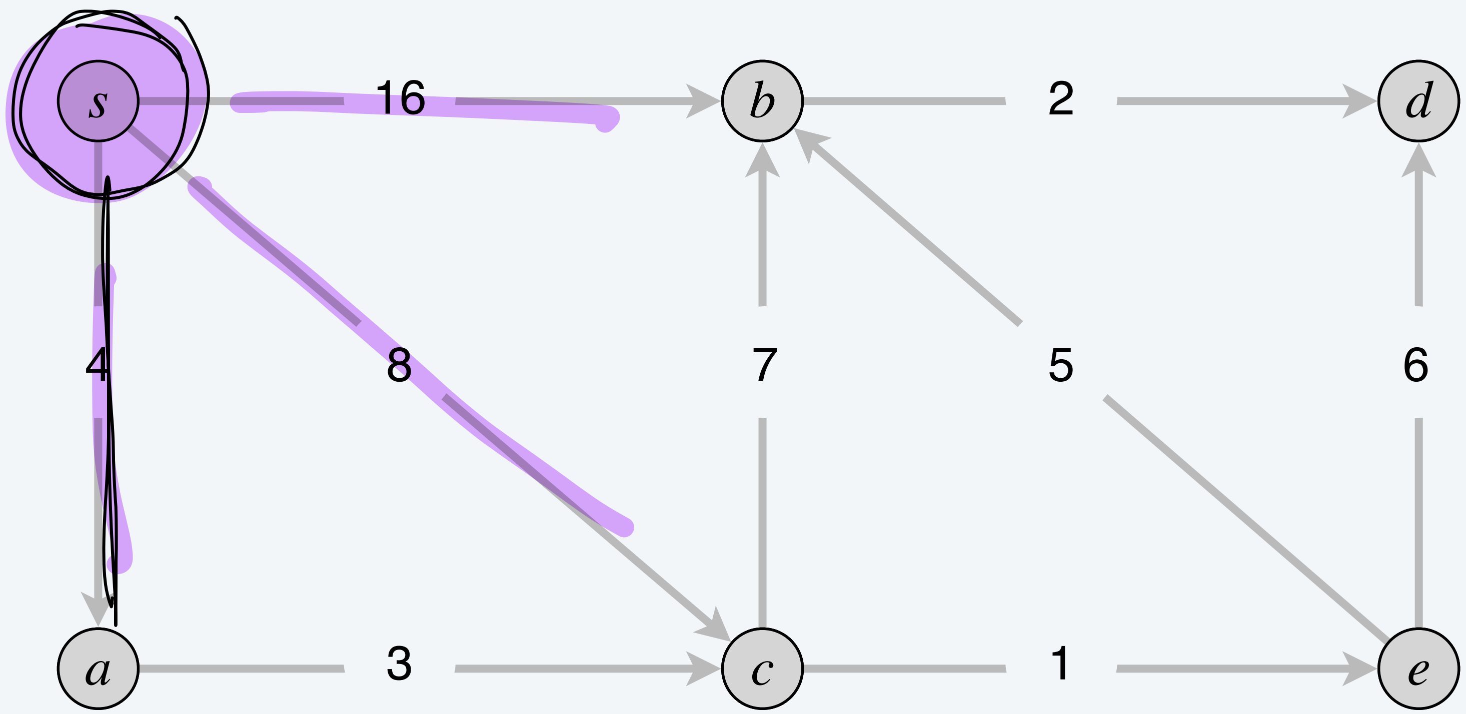
$\in$

$d'(v_2) =$

Trace through the algorithm with your table

find $v \notin S$ s.t.

$$d'(v) = \min_{e=(u,v):u\in S} \left[d(u) + \ell_e \right]$$ is smallest.



$d'(a) = \min(0+4)$
$= 4$

$d'(b) = \min(0+16)$
$= 16$

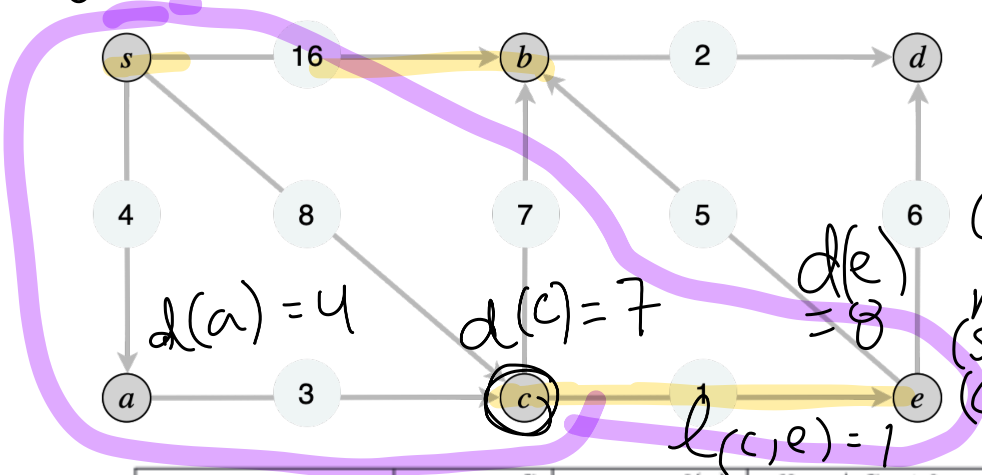
$d'(c) = \min(0+8)$
$= 8$

	current S	current $d(u)$ values for $u \in S$	all $v \notin S$ with at least one edge from S	values of $d'(v)$	v to add to S
set up	n/a	n/a	n/a	n/a	s
while loop run 1	$\{s\}$	$d(s) = 0$	b, c, a		a
while loop run 2	$\{s, a\}$				

$S = \{s, a, c\}$

$d(s) = 0$

$d'(e) = \min(d(c) + 1) = \min(7+1) = 8$



$d'(b) = \min(d(s) + l_{(s,b)}, d(c) + l_{(c,b)})$

$\min(0+16, 7+7) = 14$

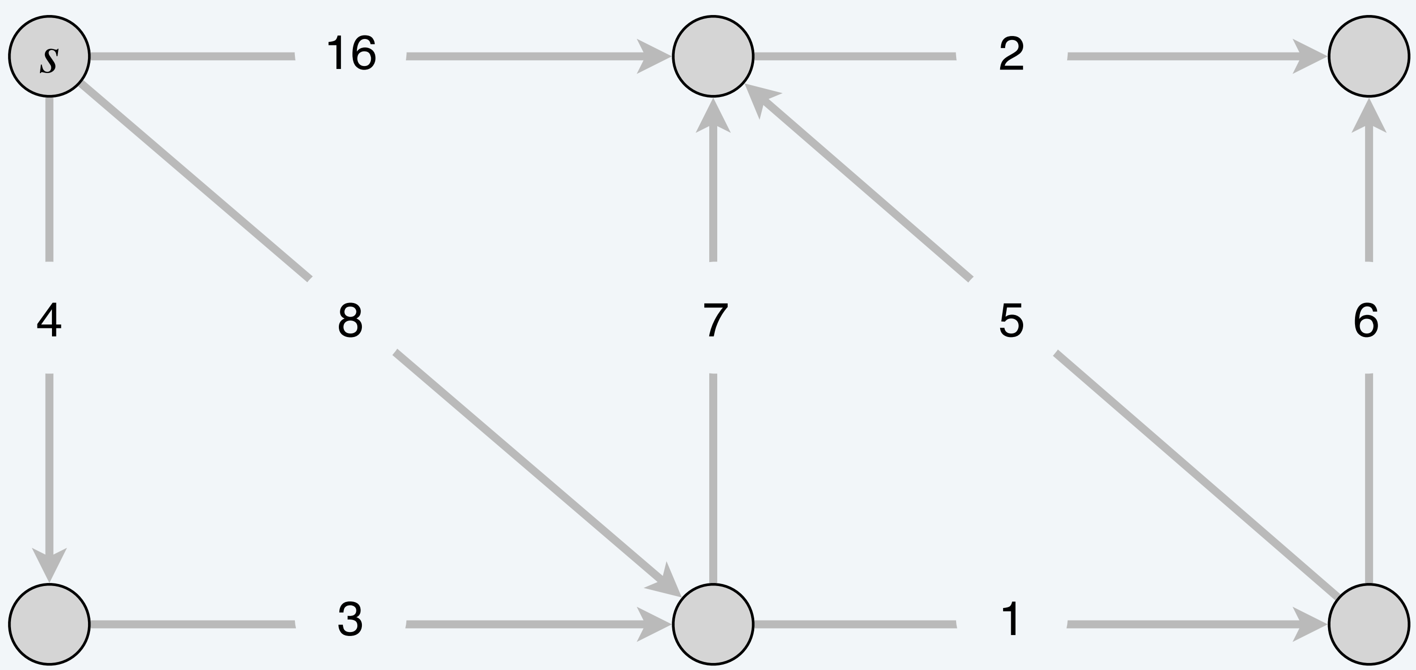
	current S	current $d(u)$ values for $u \in S$	all $v \notin S$ with at least one edge from S	values of $d'(v)$	v to add to S
set up					
while loop run 1					
while loop run 2					
while loop run 3	$S = \{s, a, d\}$	$d(s)=0$ $d(a)=4$ $d(c)=7$	e, b	$d'(e) = 8$ $d'(b) = 14$	e
while loop run 4					
while loop run					
while loop run					

Would Dijkstra still work if we didn't *greedily* choose the next node?

With your table, find a counterexample (incorrect d[u] labeling)

- Repeatedly choose unexplored node $v \notin S$ and add it to S with

$$d[v] = \min_{e=(u,v):u \in S} d[v] + \ell_e$$



Dijkstra's algorithm: proof of correctness

Dijkstra's algorithm: proof of correctness

Theorem: Every Y has quality Z.

Let x be an arbitrary Y.

Suppose that for all w less than x, quality Z holds.

There are (at least two) cases:

Case 1: non inductive case, aka base case. can prove directly that theorem holds.

(but there could be more than one of these!)

Case 2: inductive case. need to use inductive hypothesis to show that theorem holds. (but there could be more than one of these!)

x has quality Z. Because x was an arbitrary Y, every Y has quality Z.

Dijkstra′s algorithm: proof of correctness

break?

10:25 return

FlatRate

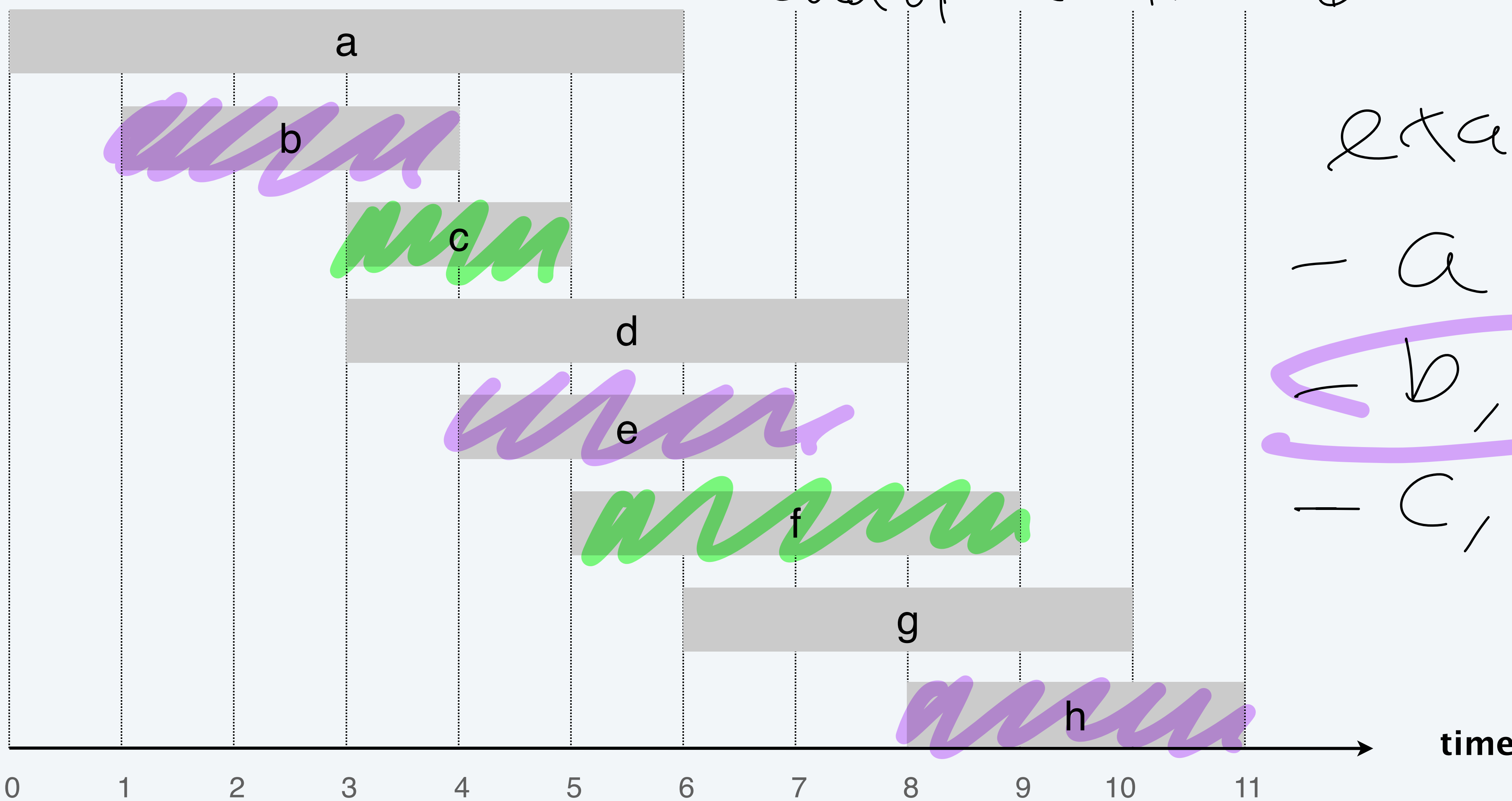
each w/ specific start, end time

- To work for FlatRate, you sign up for a shift and are given a large set of possible jobs to complete during the shift. Some jobs take longer than others, but all pay the same. Before your shift, you select which jobs to take.
- Two jobs are compatible if they don't overlap.

FlatRate

- To work for FlatRate, you sign up for a shift and are given a large set of possible jobs to complete during the shift. Some jobs take longer than others, but all pay the same. Before your shift, you select which jobs to take.
- Two jobs are compatible if they don't overlap.
- With table: what is the highest-paying set of compatible jobs for this schedule?

start of a time 0
end of a time 6

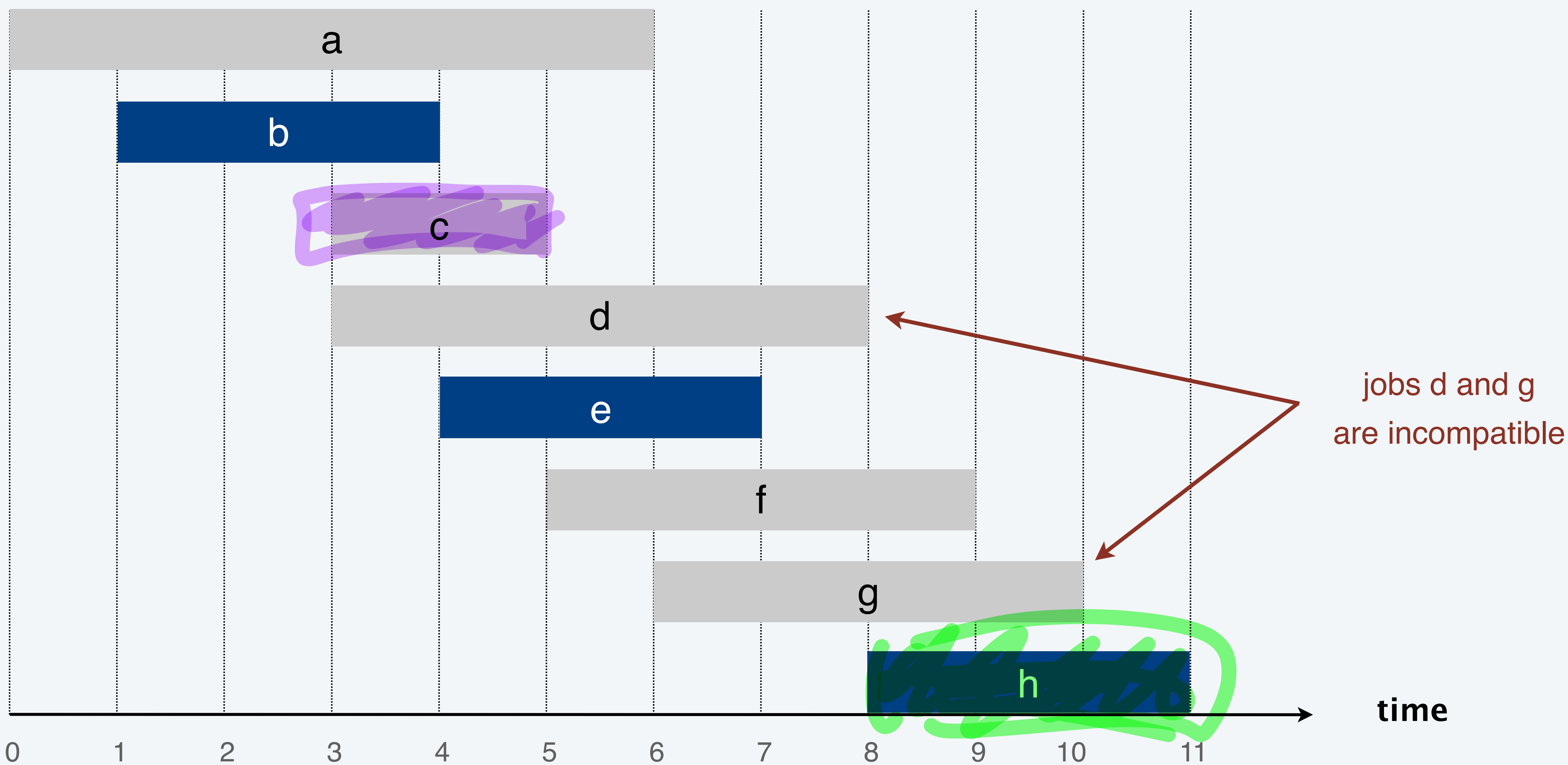


example:
- a and h
- b, e, h
- c, f

Interval scheduling

- Job j starts at s_j and finishes at f_j.
- Two jobs are compatible if they don't overlap.
- Goal: find maximum subset of mutually compatible jobs.

what set of jobs do we pick if I greedily select shortest jobs ?



idea for a greed algorithm:

- choose jobs in order of shortness, $f_i - S_i$

Some local criteria that won't work...

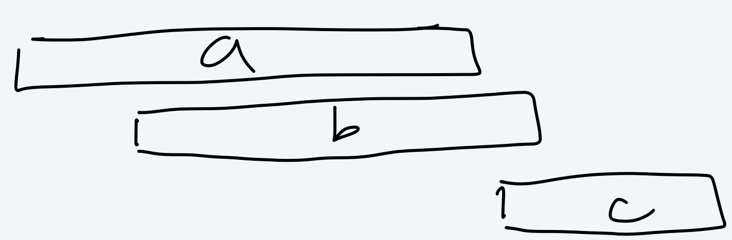
Give a counterexample to why a greedy algorithm using each of the following local criteria would yield a global optimal solution for every input.

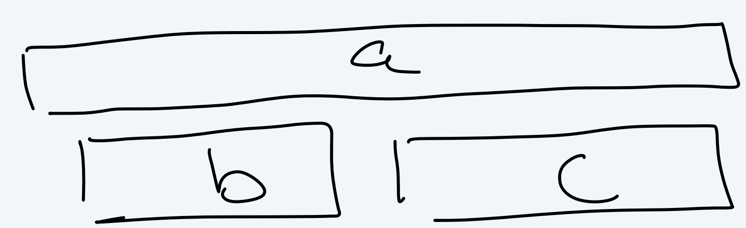
A. [Earliest start time] Consider jobs in ascending order of s_j.

B. [Shortest interval] Consider jobs in ascending order of $f_j - s_j$.

C. [Fewest conflicts] Consider jobs in ascending order of conflicts.

A: not a counterexample





a, c

a – one job

b, c – two jobs

go over quiz 4