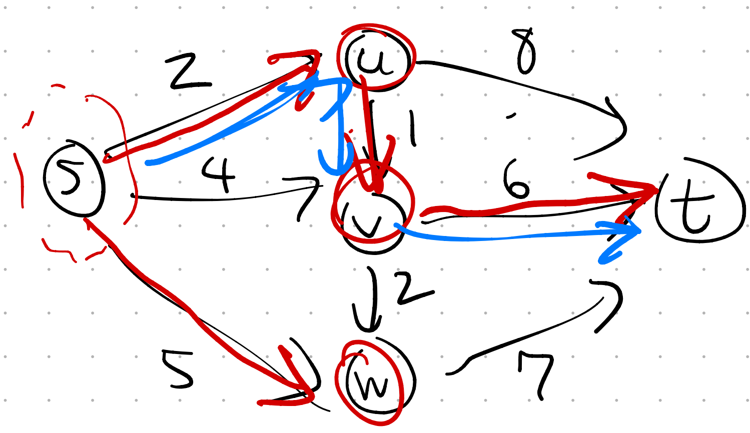
7 Feb 2025 Shortest Paths and Bellman–Ford

(Guest lecture: Bobby Kleinberg)

In an <u>unweighted</u> graph, shortest s-t path is found by BFS (breadth-first search) in $O(m+n)$ time.

#edges #vertices

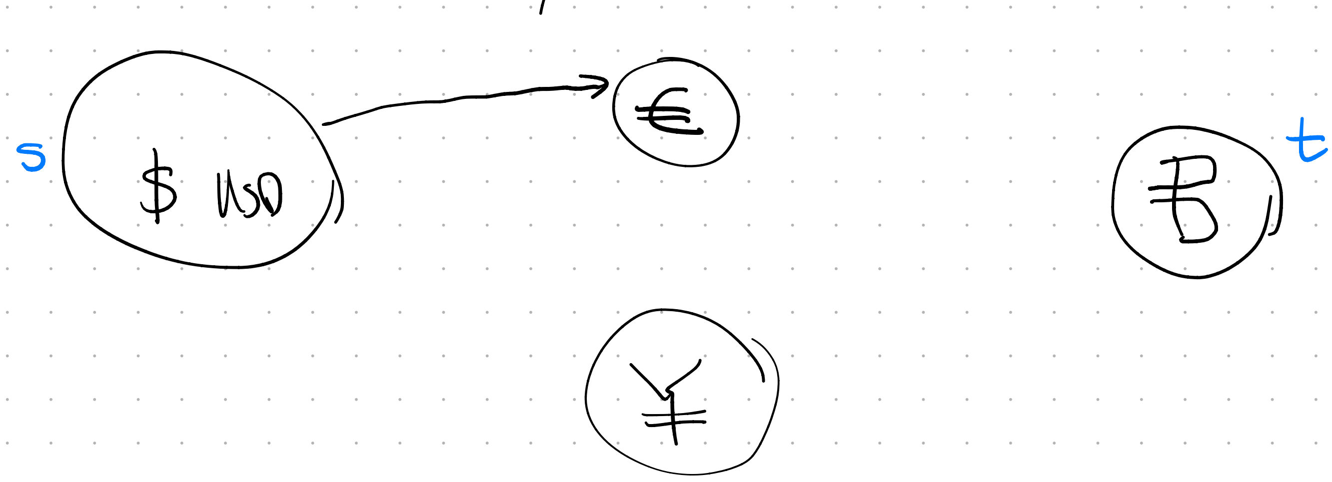
If edges have <u>costs</u> $c(u,v) \geq 0$ and we're searching for s-t path on <u>minimum</u> <u>total</u> <u>cost</u>, we can use <u>Dijkstra's</u> <u>Algorithm</u> in $O(m \log n)$ time.



Today's algorithm (Bellman–Ford) finds a min-cost s-t path in graphs that may have positive and negative edge costs.

(BUT NO NEGATIVE COST CYCLES)

<u>Ex. 1</u>. Currency exchange



For edge u,v say $r(u,v)$ denotes
of u units needed to buy <u>one</u> unit of v.

<u>Problem.</u> What's the minimum # of s units to obtain 1 unit of t through a sequence of exchanges?

$$s = u_0 \longrightarrow u_1 \longrightarrow u_2 \longrightarrow \cdots \longrightarrow u_k = t$$

To get 1 unit of t you need to start with

$$r(u_0,u_1) \cdot r(u_1,u_2) \cdot \;\cdots\; \cdot r(u_{k-1},u_k)$$

units of s.

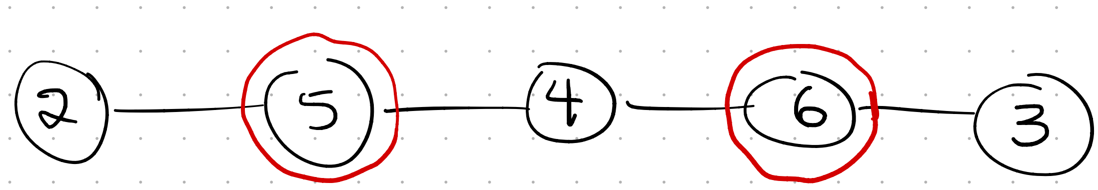
Goal is to find a path that minimizes this product.

Equivalently, minimize

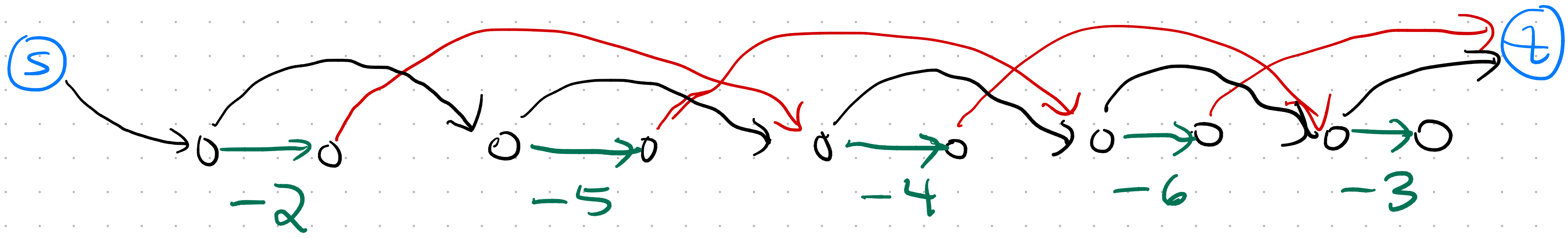
$$\sum_{i=1}^{k} \log(r(u_{i-1},u_i))$$

define to be $c(u_{i-1},u_i)$

Recall WIS:



Given path labeled with numbers on nodes, find a set of pairwise non-adjacent nodes, that maximizes sum of labels.



Black and red edges have zero cost.

Paths in this directed graph and independent sets in the original problem are in bijective correspondence.

Under this bijection

$$\text{weight}(\text{ind set}) = -\text{cost}(\text{path})$$

In other words we have described a reduction from WIS to shortest path.

To solve shortest path from s to t we generalize the problem to:

$\forall$ vertex v, $\forall$ hop-count h find min-cost path from s to v composed of h or fewer edges.

If $h=0$, the only v reachable from s in h or fewer hops is s itself.

If $h>0$, a path of h or fewer hops from s to v is either:

(a) a path of $h-1$ or fewer hops from s to v, or

(b) a path of $\leq h-1$ hops from s to u followed by an edge (u,v).

(R) $$\text{MIN-COST}(s,v,h) = \min \begin{cases} \text{MIN-COST}(s,v,h-1), \\ \min\limits_{e=(u,v)} \text{MIN-COST}(s,u,h-1) + \text{cost}(u,v) \end{cases}$$

(I) $$\text{MIN\_COST}(s,v,\phi) = \begin{cases} 0 & \text{if } s=v \\ \infty & \text{if } s\neq v \end{cases}$$

for $h=0,1,2,\ldots,n-1$:
  for $v\in V$:
    if $h=0$ fill in MIN COST(s,v,h) using (I)
    if $h>0$ " " " " using (R)
output MIN-COST$(s,t,n-1)$

RUNS IN $O(mn)$ time.