Feb 23, 2016

Reminder: Exam 1 on Thursday

Exam 1

(1) A double-sided handwritten "cheaty" sheet

US-letter sheet

(2) Calculator

5 Problems

P1 sorting: any linear order can be sorted

P2 Basic data struc

P3 Modify some known data structure

P4 Special type of sorting problem

$O(n)$ numbers k distinct numbers $k \leq n$

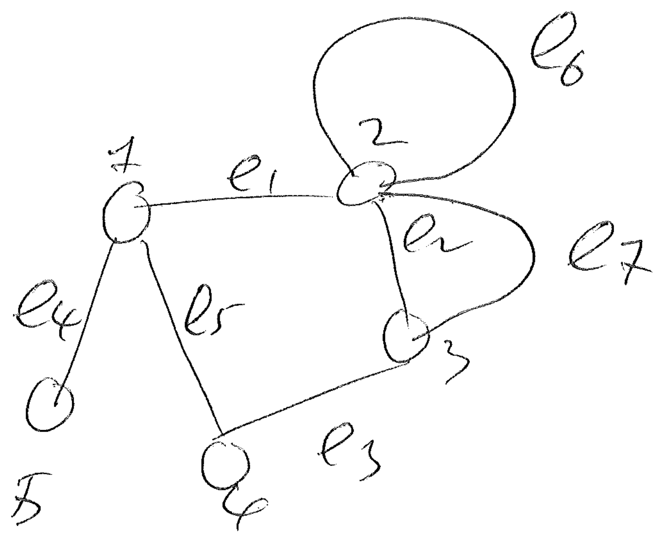
$n \lg k$

P5.

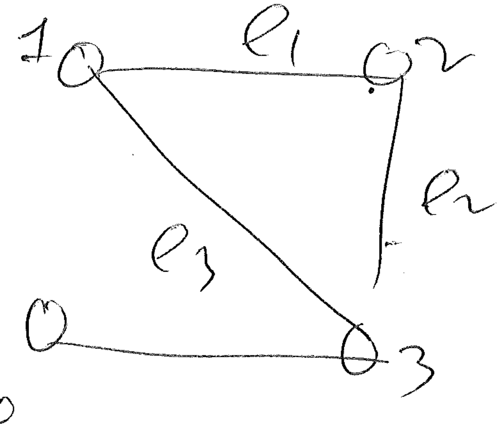
12:20pm — 1:50pm

Graph Algorithms

A graph

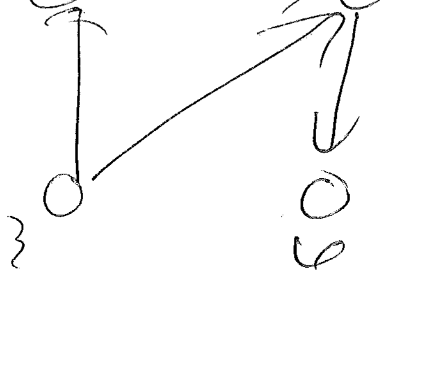


Simple Graph : No self loops or multi edges



undirected tree : a connected graph with no cycles

Directed Graph



Directed Graph (Digraph) $\equiv$

Relation $\equiv$

Boolean Matrix

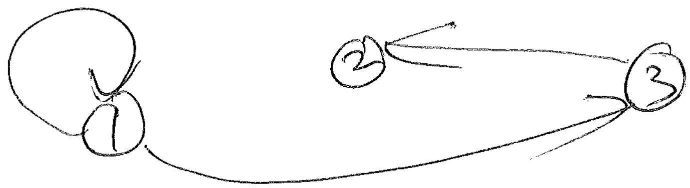
Recall a relation ~~R~~ E on a set V is a subset of $V \times V$

Recall Boolean Matrix 0-1 matrix

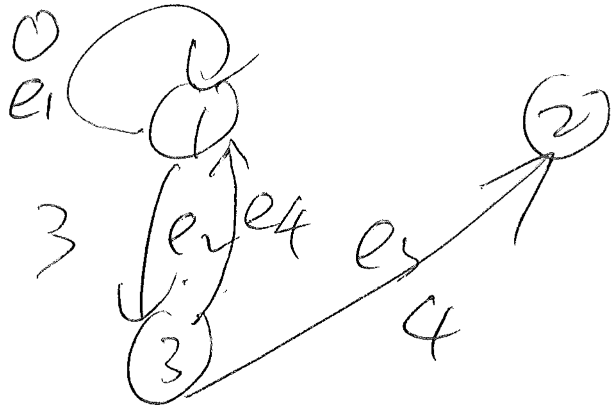
$V = \{1, 2, 3\}$

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

$$B_E = \begin{array}{c|ccc} & 1 & 2 & 3 \\ \hline 1 & \\ 2 & \\ 3 & \end{array} \begin{bmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$ ← adjacency matrix



Weighted Graph

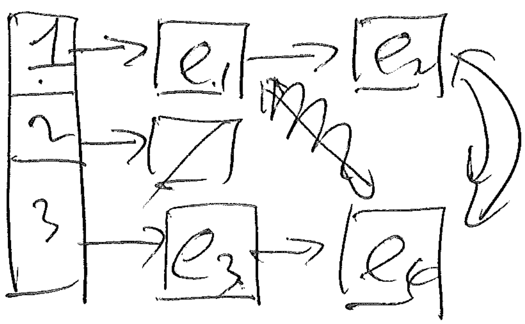


$$
\begin{array}{c|ccc}
 & 1 & 2 & 3 \\
\hline
1 & 0 & \infty & 3 \\
2 & \infty & \infty & \infty \\
3 & \infty & 4 & \infty
\end{array}
$$

Sparce Graph

$$|E| = o(|V|^2)$$

Adjacency List

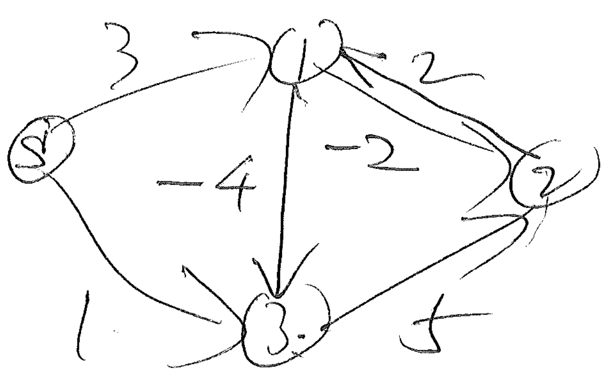


$$O(|V| + |E|)$$

Shortest Paths on General Graph

Single source shortest path:

Given a weighted graph $G(V,E)$ and a source vertex $s \in V$, find the shortest path from s to all other vertices



What if there are negative edges?

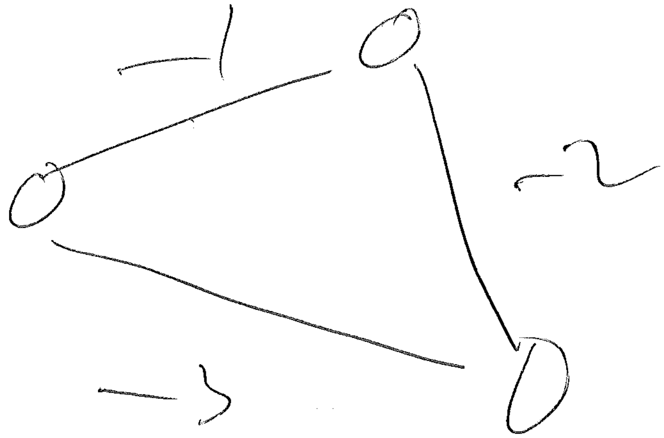
When there are negative cycles in the graph, the shortest path may not be well-defined.

The algorithm should determine if the given graph has a negative cycle or not

If yes, terminate

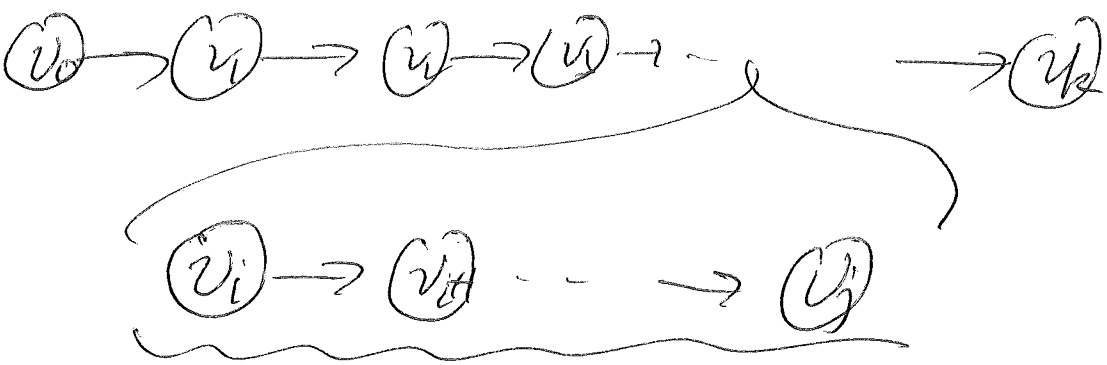
If no, calculate the shortest paths

Digression



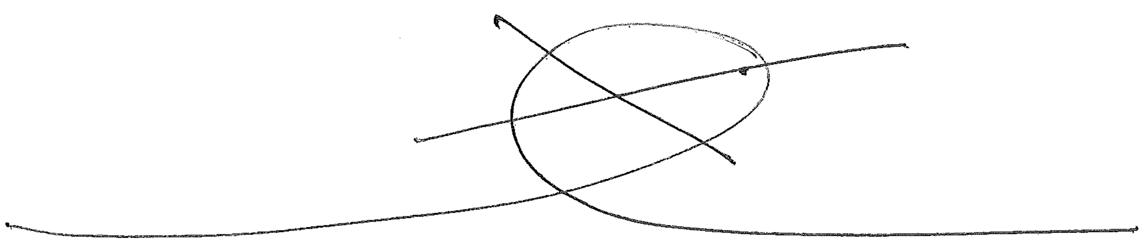
Suppose the shortest path are well-defined.

(1) All subpaths of a shortest path are also shortest paths



a shortest path from v_i to v_j

(2) There are no cycles in the shortest path

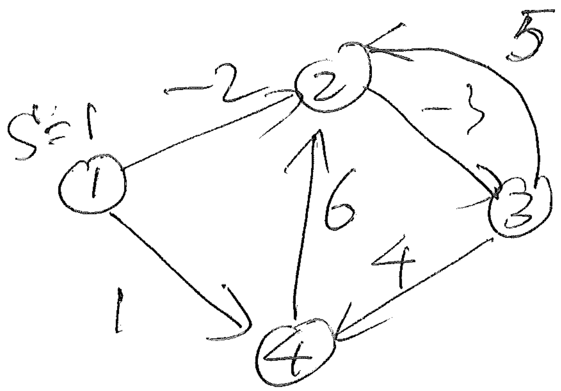


Bellman - Ford Algorithm for single source shortest paths

induction on the number of edges on the shortest path

In each iteration, the algorithm calculates the shortest paths from the source to all other vertices using $\leq k$ edges.

Basis $k=0$

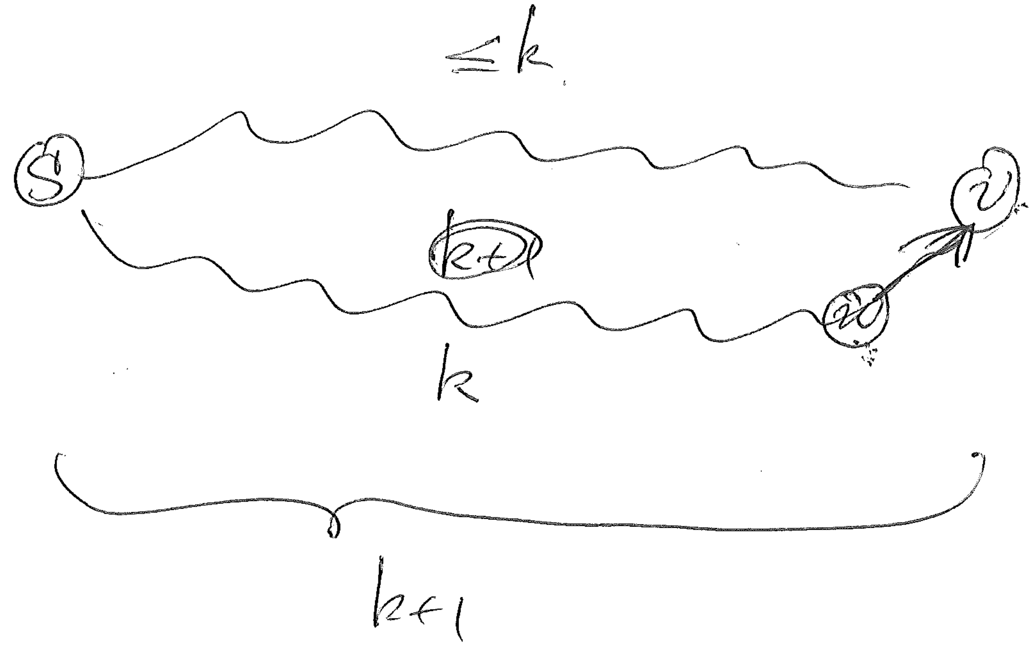


Basis the shortest paths from vertex 1 to all other vertices using ≤ 0 edges.

1	2	3	4
0	∞	∞	∞

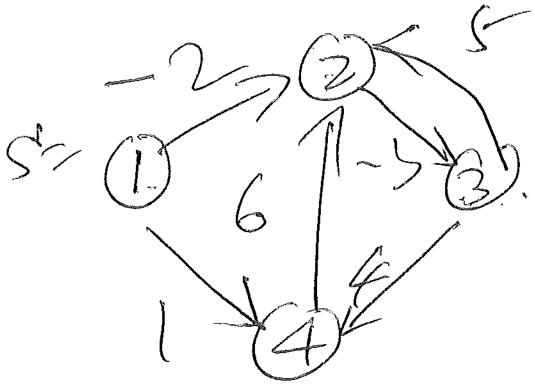
I.H., assume we know the shortest path from the source to all other ~~over~~ vertices using $\leq k$ edges

Can we find the shortest path using $\leq k+1$ edges?



~~SP[v][k+1]~~

$$SP[k+1][v] = \min \begin{cases} SP[k][w] + c(w,v) \\ \text{forall } w \neq v, s \\ \\ SP[k][v] \end{cases}$$



	1	2	3	4
1	0	-2	∞	1
2	∞	0	-3	∞
3	∞	5	0	4
4	∞	6	∞	0

k	1	2	3	4
0	0	∞	∞	∞
1	0	-2	∞	1
2			?	
3				

shorter path fm 1 to 3 usig ≤ 2 edges

$$SP[2][3] = \min \left\{ \begin{array}{l} SP[1][3] + C[3,3] \\ SP[1][2] + C(2,3) \\ SP[1][4] + C(4,3) \\ \cancel{SP[1][1] +} \\ SP[1][1] + C(1,3) \end{array} \right\}$$

Bellman-Ford.

input : adjacency matrix $C_{n \times n}$, $s = 1$

output: $SP_{n \times n}$

Basis. ~~SP[0][0] =~~

$SP[0][1] = 0 \qquad SP[0][j] = \infty, \ j \neq 1$

$SP[1][*] = C[1][*];$

For $k = 2$ to $n-1$

For $j = 1$ to n

$$SP[k][j] = \min \left\{ \begin{array}{l} SP[k-1][1] + C[1][j] \\ SP[k-1][2] + C[2][j] \\ \vdots \\ SP[k-1][n] + C[n][j] \end{array} \right\}$$

	1	2	3	4
1	0	-2	∞	1
2	∞	0	-3	∞
3	∞	5	0	4
4	∞	6	∞	0

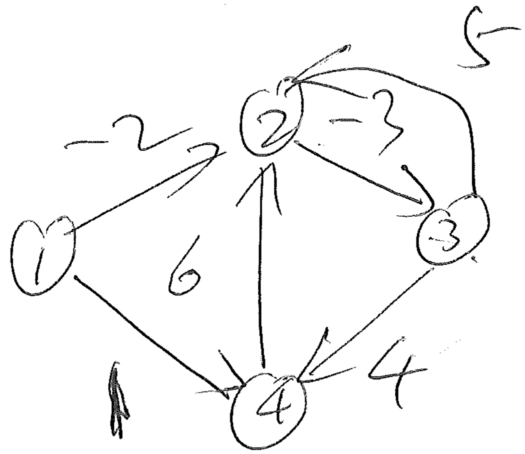
k	1	2	3	4
1	0	-2	∞	1
2	0	-2	-5	
3				

$s = 1$

$$(0, -2, \infty, 1) \ \text{min+} \begin{pmatrix} 0 \\ \infty \\ \infty \\ \infty \end{pmatrix}$$

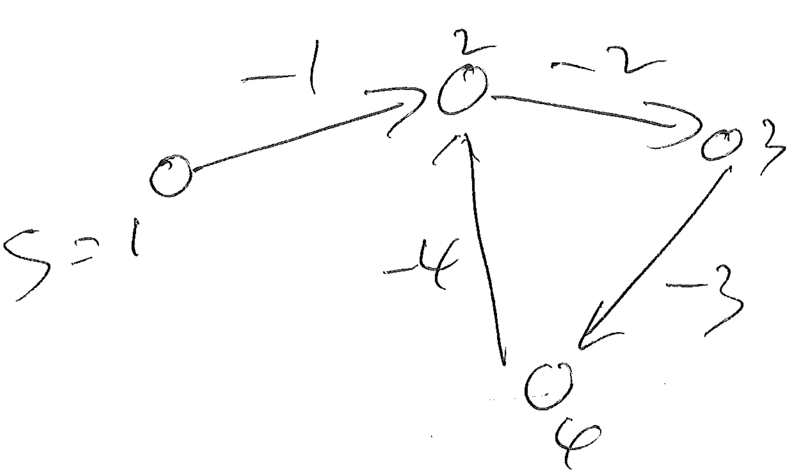
$$(0, -2, \infty, 1) \ \text{min+} \begin{pmatrix} -2 \\ 0 \\ 5 \\ 6 \end{pmatrix}$$

$$(0, -2, \infty, 1) \begin{pmatrix} \infty \\ -3 \\ 0 \\ \infty \end{pmatrix}$$



The order the shortest paths are discovered.

(1,1)	0	0 hops
(1,2)	-2	1 hops
(1,3)	-5	2 hops
(1,4)	-1	3 hops



k	4
1	∞
2	∞
3	-6
4	-6
5	-6
6	-16