Graphs

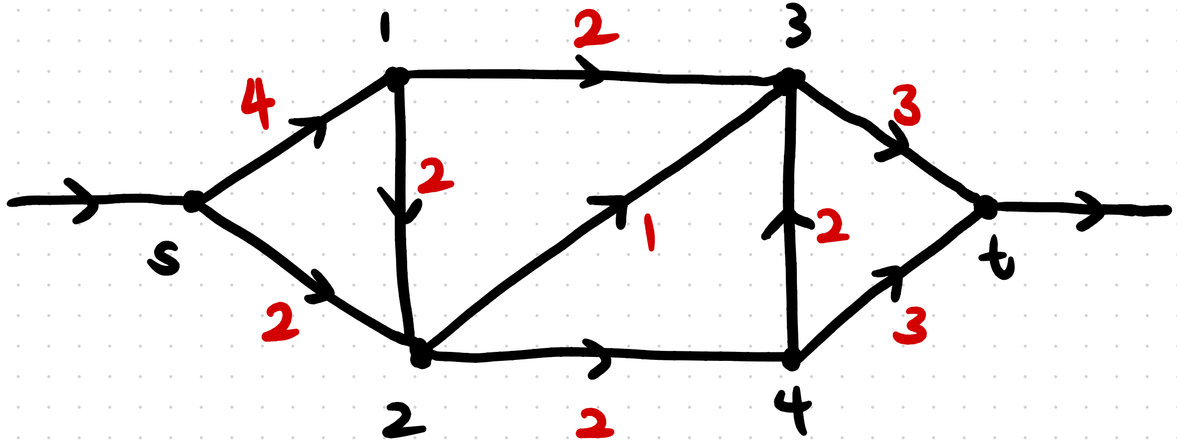
cont.

(8.3)

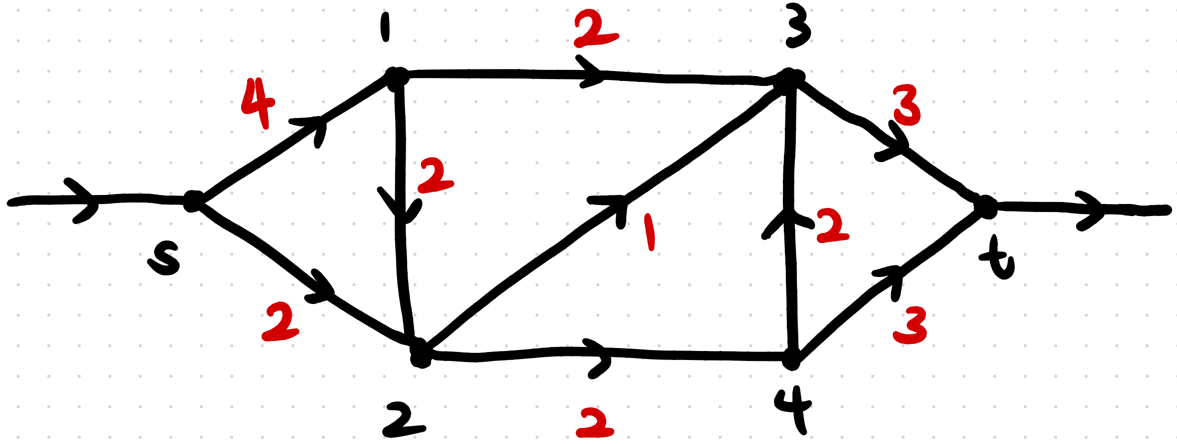
We have n workers and n tasks that need completing. Each worker has a different set of jobs they can perform. We want to assign tasks so that each worker does exactly one task, and every task has a worker assigned.

What if some workers can do some tasks more efficiently than others?

Oil is being shipped from point s to point t through the following network of pipes. The pipes have different diameters, and the maximum amount of oil per hour each pipe can transport is shown. How much oil per hour can be transported from s to t?



Oil is being shipped from point s to point t through the following network of pipes. The pipes have different diameters, and the maximum amount of oil per hour each pipe can transport is shown. How much oil per hour can be transported from s to t?



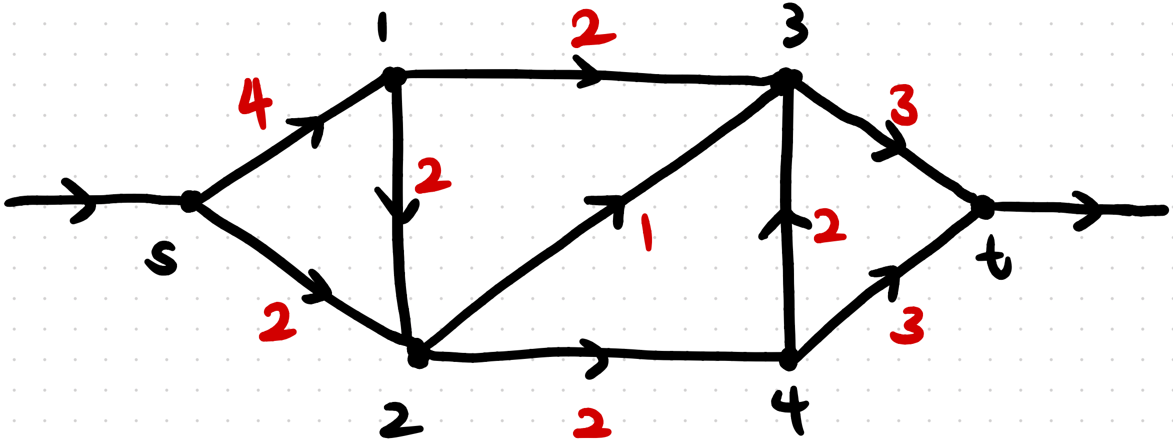
How are graphs implemented in computer programs?

1) Adjacency matrix

Matrix whose rows and columns are indexed by vertices, and

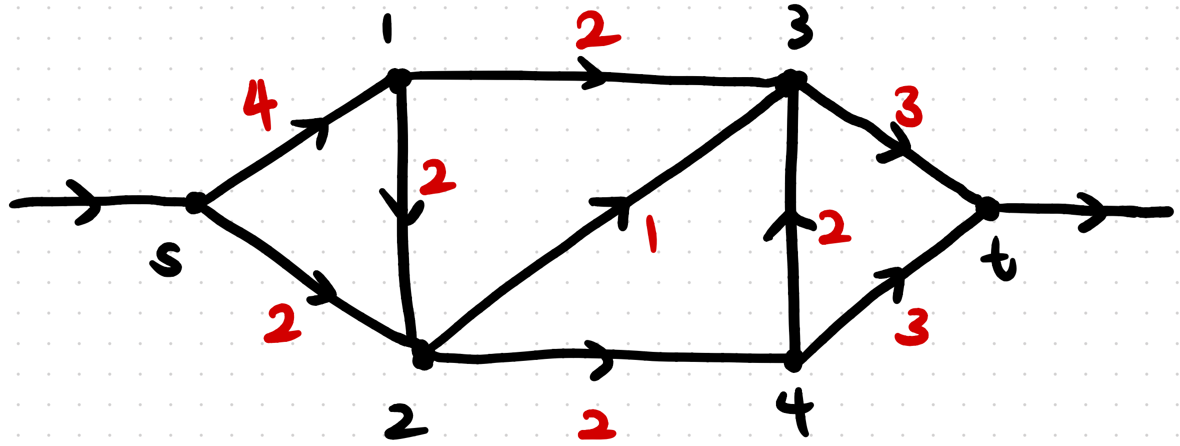
$$a_{ij} = \begin{cases} 1 & \text{if } i \text{ and } j \text{ are adjacent} \\ 0 & \text{otherwise} \end{cases}$$ for undirected graphs

$$a_{ij} = \begin{cases} 1 & \text{if } ij \text{ is an edge} \\ -1 & \text{if } ji \text{ is an edge} \\ 0 & \text{otherwise} \end{cases}$$ for directed graphs


1
2
3
4
s
2
4
2
2
1
2
3
3
t
2
4
2

2) Adjacency list

Dictionary (hash table) whose keys are vertices, and the value of each key is its set of neighbors.



Adjacency matrices are good for dense graphs (graphs with many more edges than vertices).

Adjacency lists are more common, and good for sparse graphs.