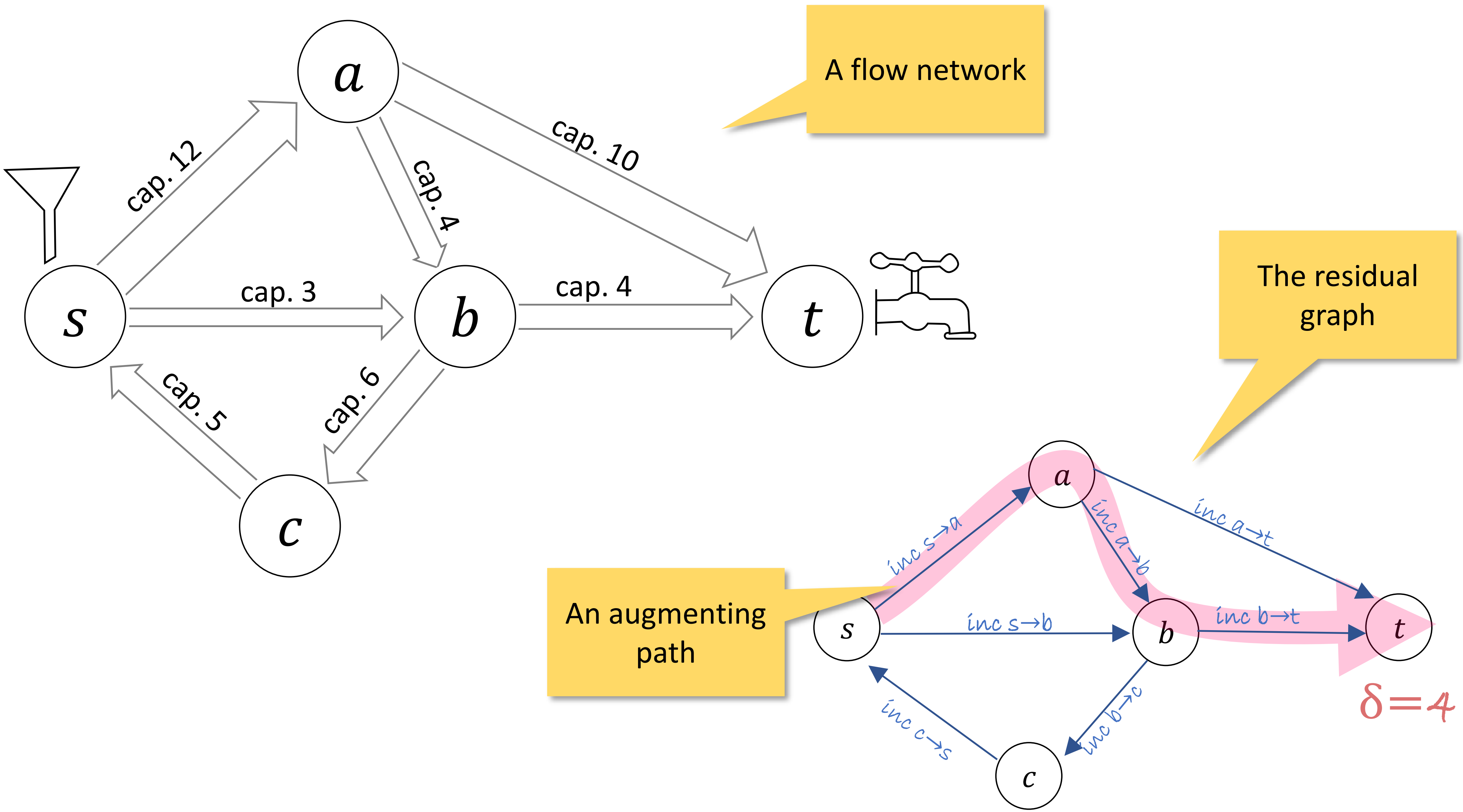
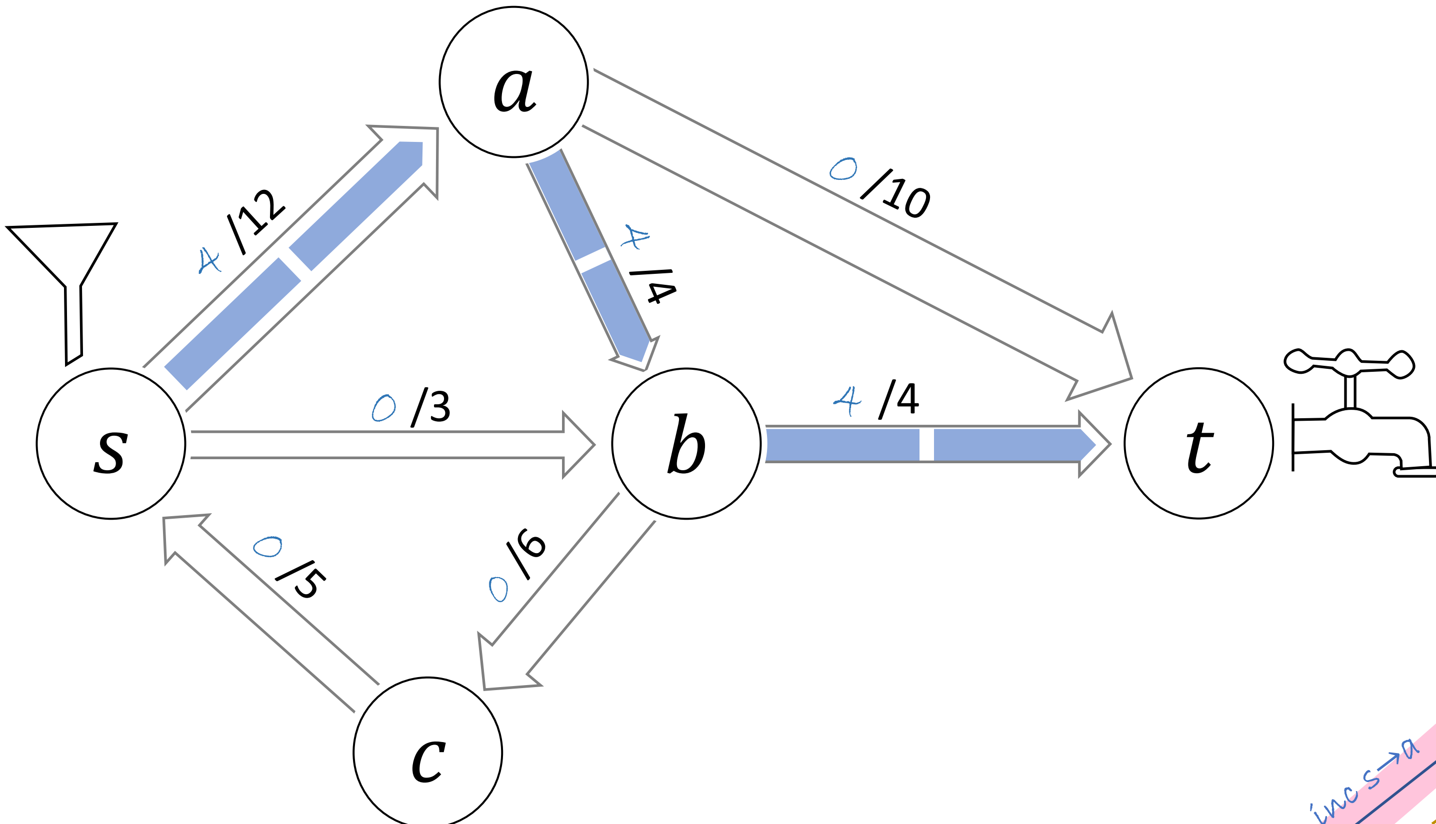
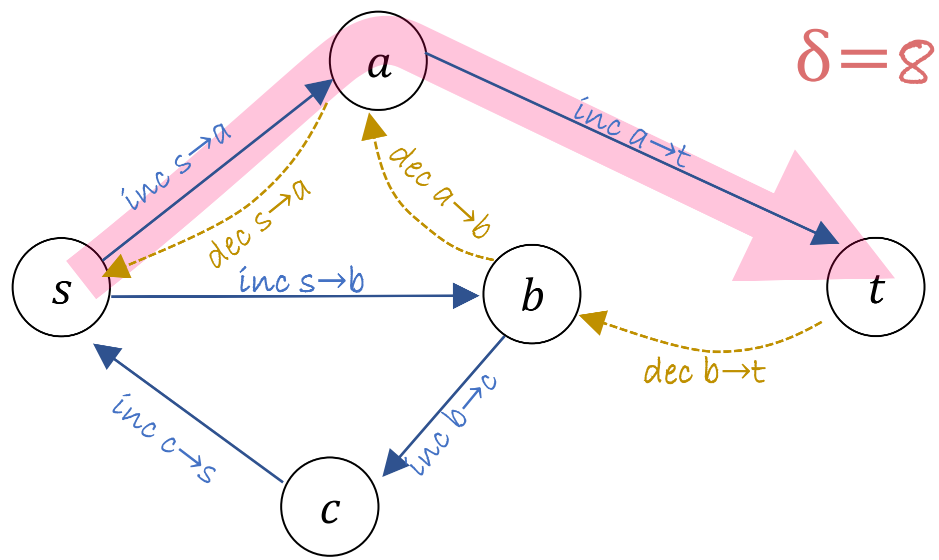
WALKTHROUGH OF FORD-FULKERSON

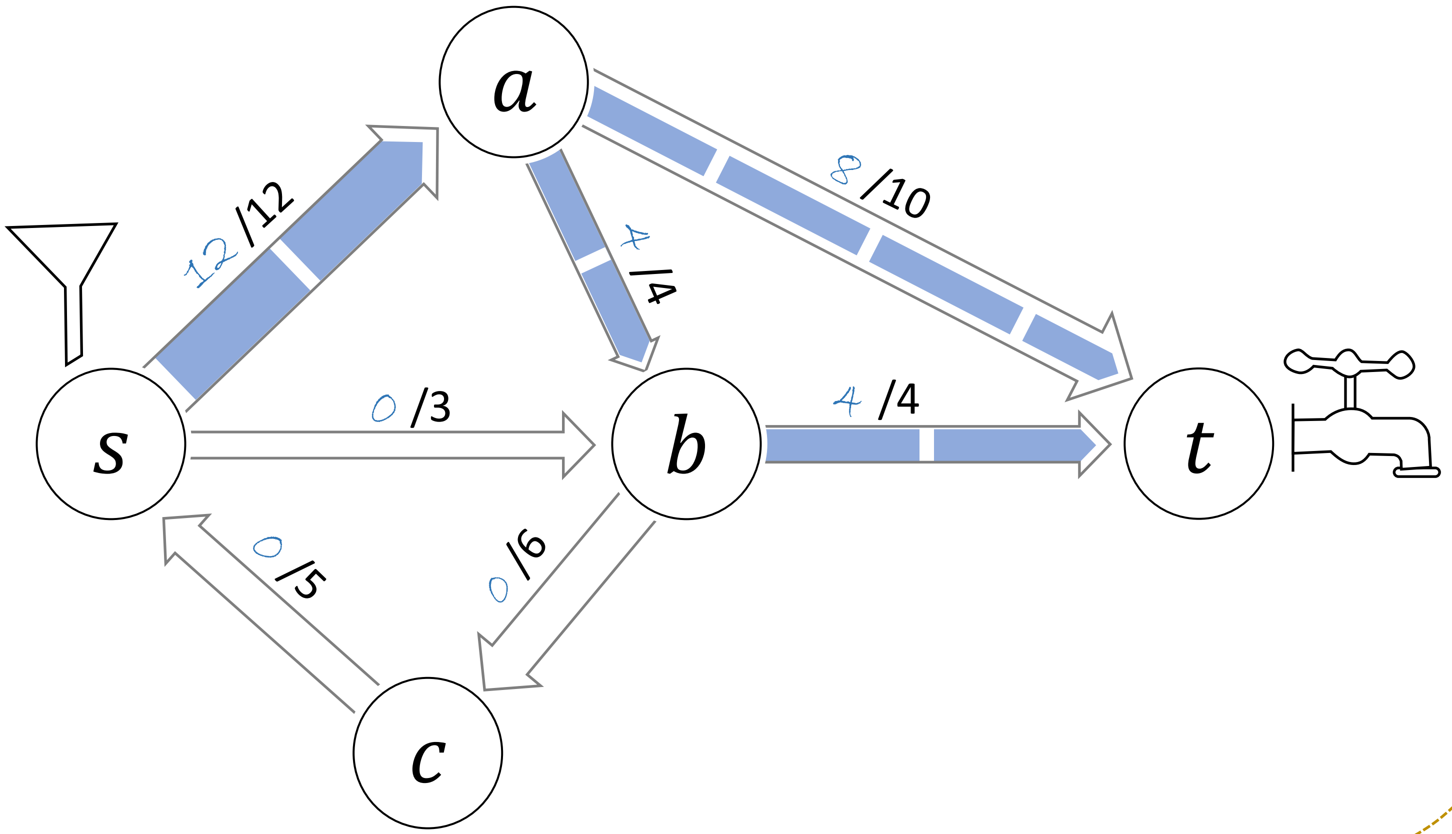


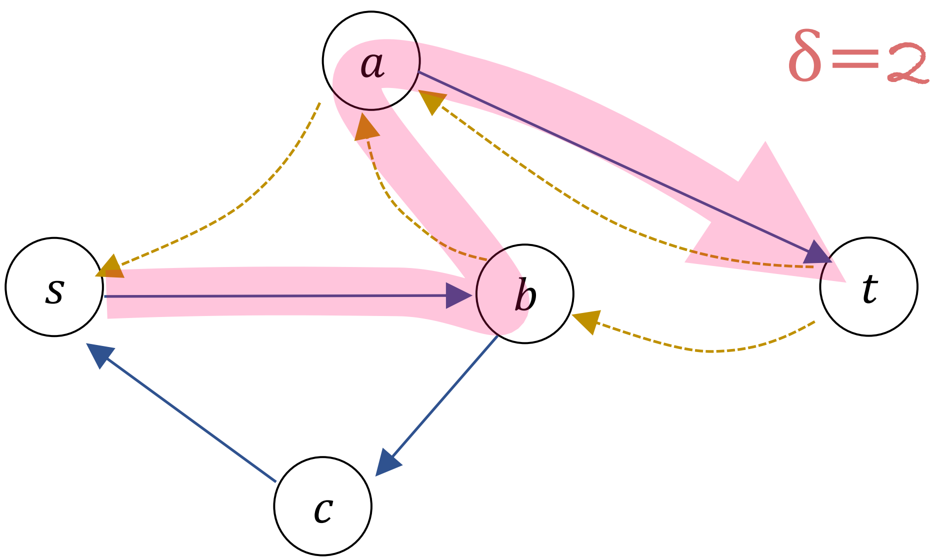
WALKTHROUGH OF FORD-FULKERSON





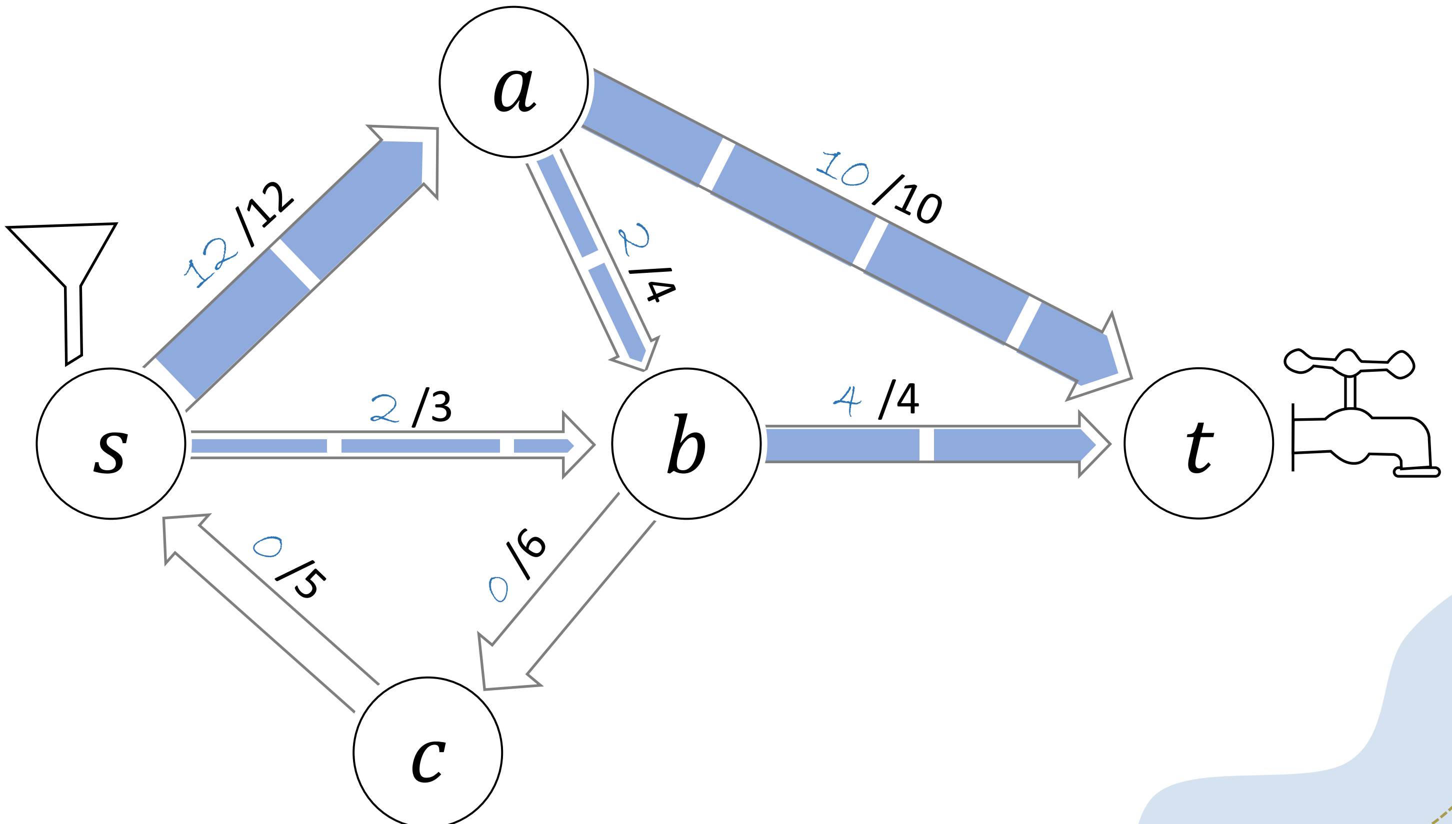
WALKTHROUGH OF FORD-FULKERSON

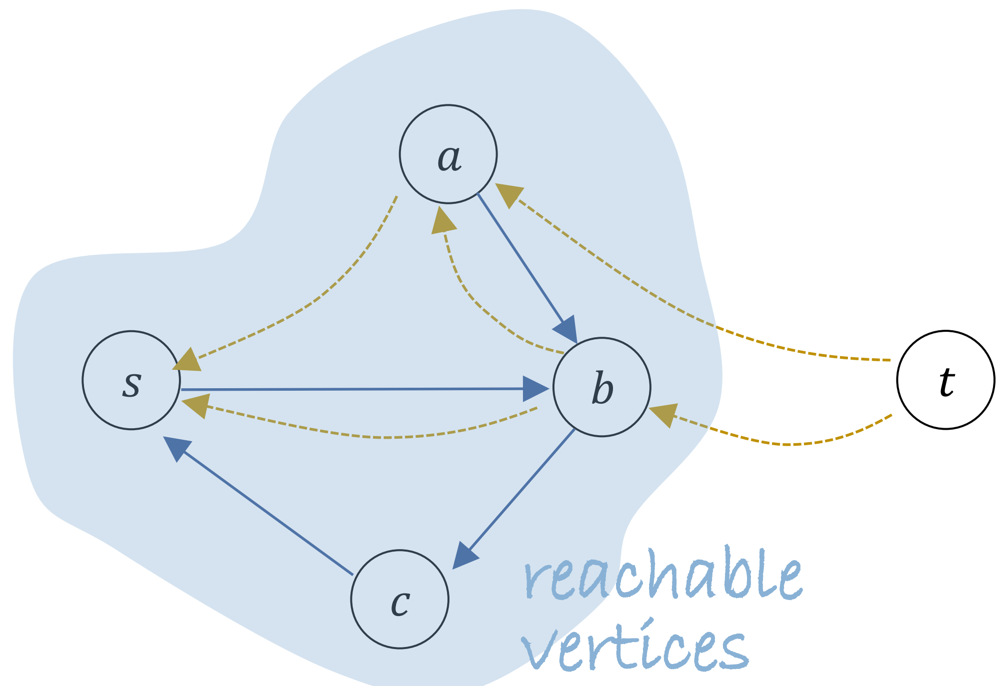




WALKTHROUGH OF FORD-FULKERSON

We cannot find an augmenting path in the residual graph. So, terminate.

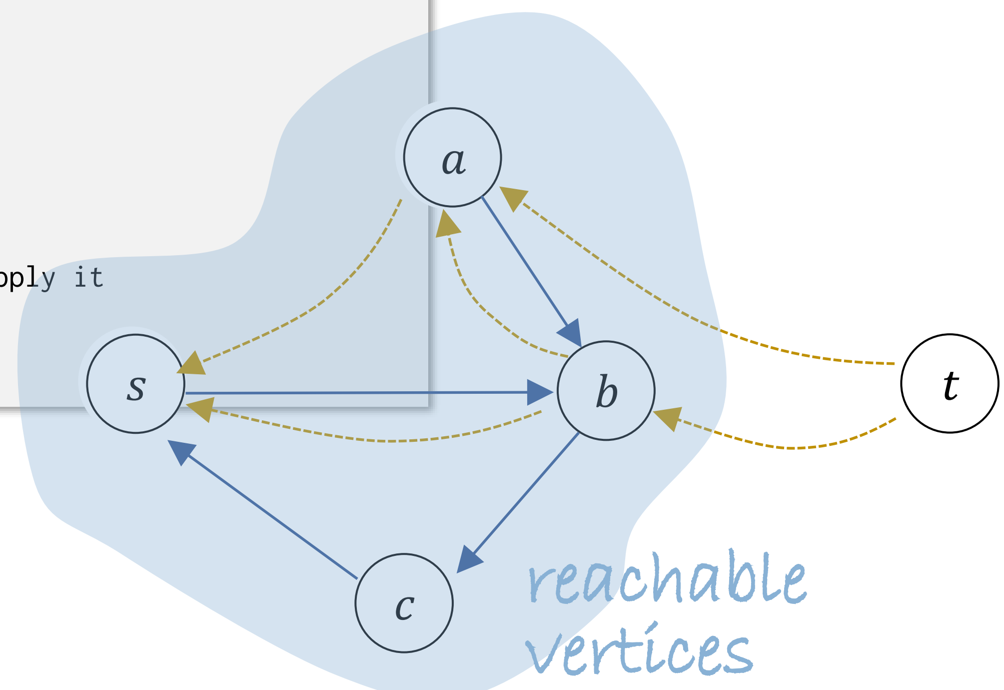




```
def ford_fulkerson(g, s, t):
    # Let f be a flow, initially empty
    for u → v in g.edges:
        f(u → v) = 0

    # Define a helper function for finding an augmenting path
    def find_augmenting_path():
        # Define the residual graph h on the same vertices as g
        for u → v in g.edges:
            if f(u → v) < c(u → v): give h an edge u → v labelled "inc u → v"
            if f(u → v) > 0: give h an edge v → u labelled "dec u → v"
        if h has a path from s to t:
            return some such path, together with the labels of its edges
        else:
            # Let S be the set of vertices reachable from s (used in the proof)
            return None

    # Repeatedly find an augmenting path and add flow to it
    while True:
        p = find_augmenting_path()
        if p is None:
            break
        else:
            compute δ, the amount of flow to apply along p, and apply it
            # Assert: δ > 0
            # Assert: f is still a valid flow
```



SECTION 6.3

Max-flow min-cut

SECRET RM-1573 10-24-55 -33-

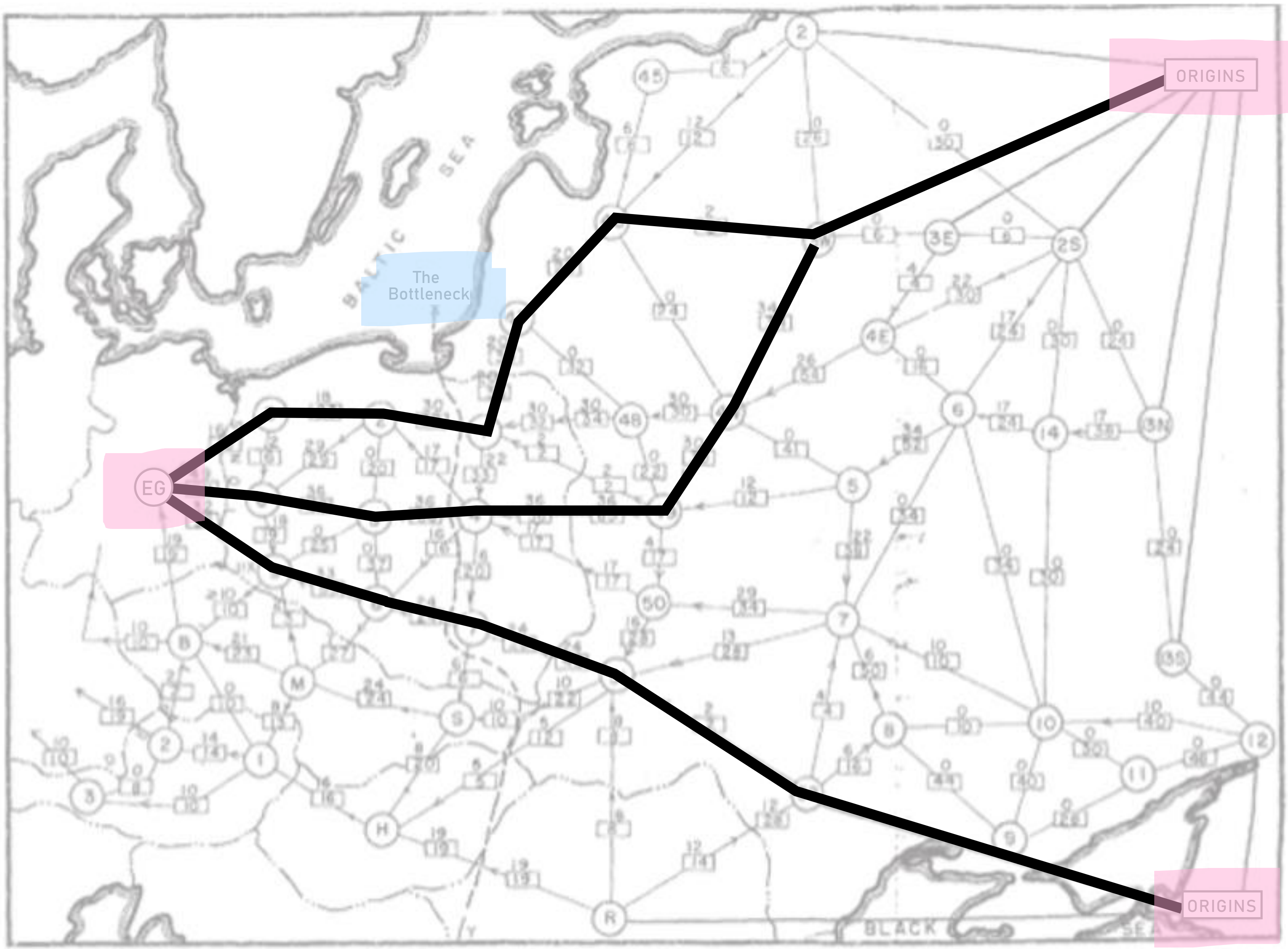


Fig. 7 — Traffic pattern: entire network available

Legend:

International boundary

Railway operating division

Capacity: 12 each way per day. Required flow of 9 per day toward destinations (in direction of arrow) with equivalent number of returning trains in opposite direction

All capacities in trains / √1000's of tons each way per day

Origins: Divisions 2, 3W, 3C, 2S, 13N, 13S, 12, 52 (USSR), and Roumania

Destinations: Divisions 3, 6, 9 (Poland); B (Czechoslovakia); and 2, 3 (Austria)

Alternative destinations: Germany or East Germany

Note IIK of Division 9, Poland

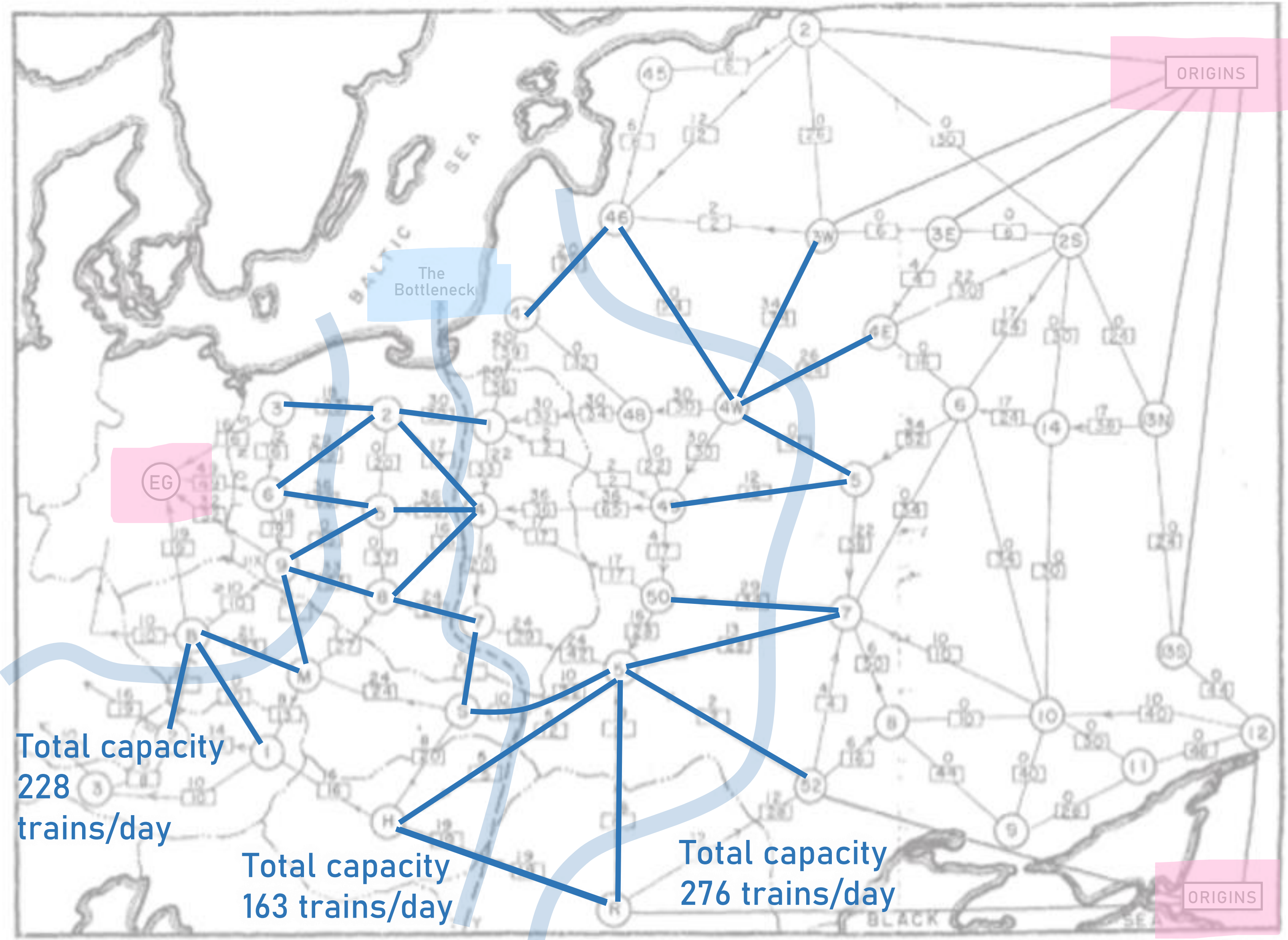


Fig. 7 — Traffic pattern: entire network available

Legend:

International boundary

Railway operating division

Capacity: 12 each way per day. Required flow of 9 per day toward destinations (in direction of arrow) with equivalent number of returning trains in opposite direction

All capacities in trains / $\sqrt{}$1000's of tons each way per day

Origins: Divisions 2, 3W, 3C, 2S, 13N, 13S, 12, 52 (USSR), and Roumania

Destinations: Divisions 3, 6, 9 (Poland); 8 (Czechoslovakia); and 2, 3 (Austria)

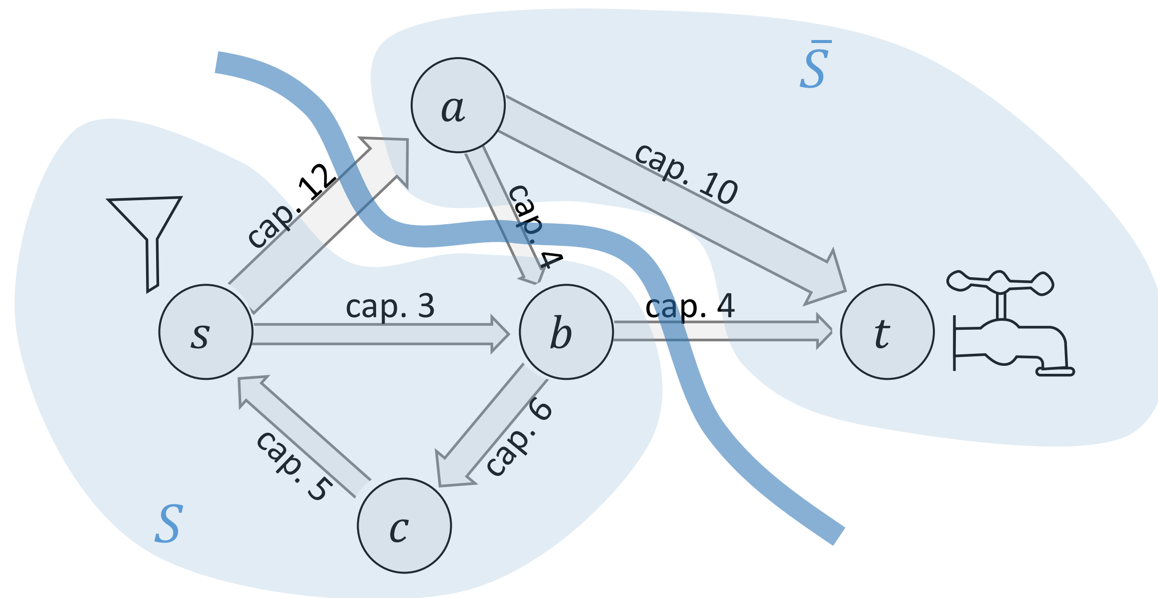
Alternative destinations: Germany or East Germany

Note IIK of Division 9, Poland

A **cut** is a partition of the vertices into two sets, $V = S \cup \bar{S}$, with the source vertex $s \in S$ and the sink vertex $t \in \bar{S}$.

The **capacity** of the cut is

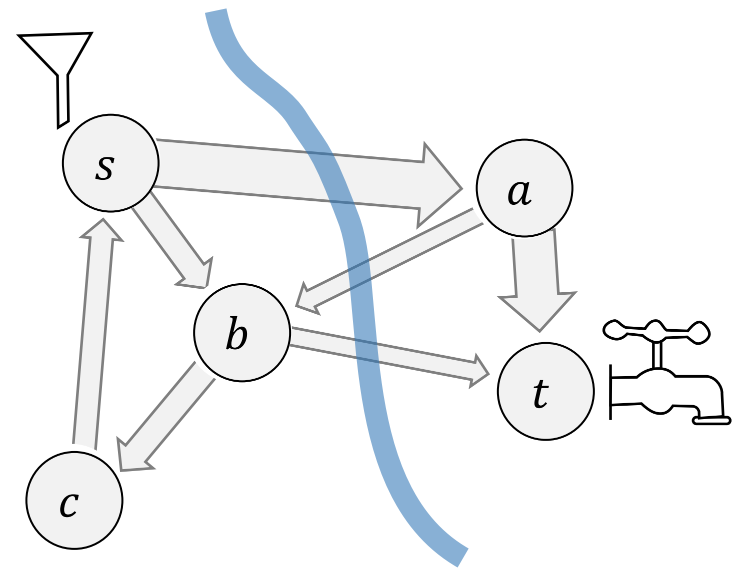
$$\text{capacity}(S,\bar{S}) = \sum_{\substack{u\in S,\, v\in \bar{S}:\\ u\to v}} c(u\to v)$$

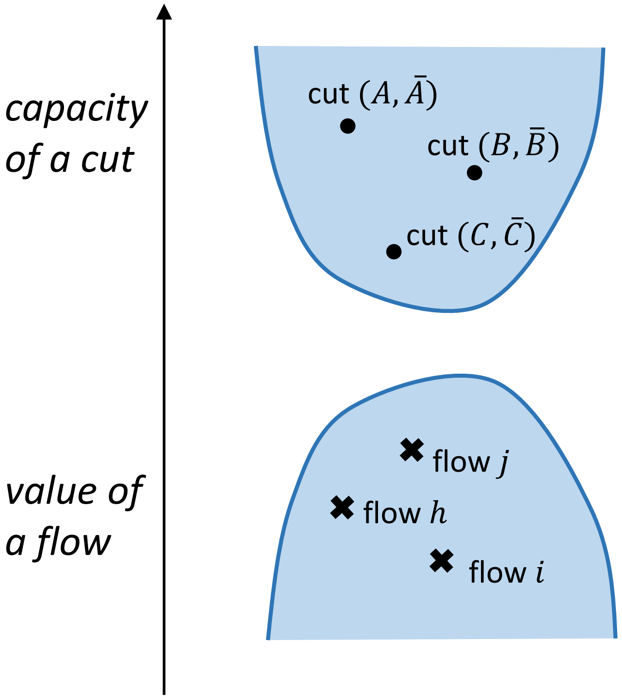


MAX-FLOW MIN-CUT THEOREM

For any flow f and any cut $(S,\bar{S})$,

$$\text{value}(f) \le \text{capacity}(S,\bar{S})$$

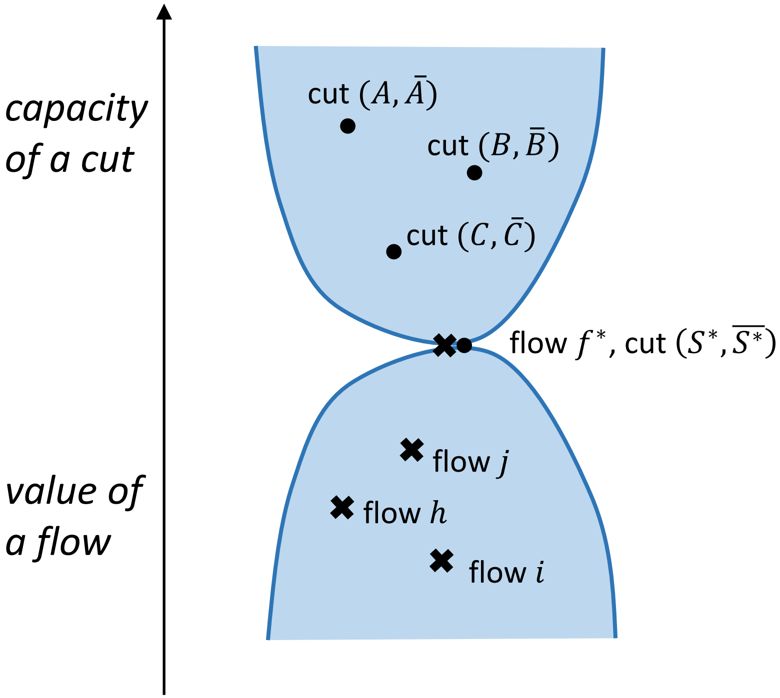




MAX-FLOW MIN-CUT THEOREM

For any flow f and any cut $(S, \bar{S})$,

$$\text{value}(f) \leq \text{capacity}(S, \bar{S})$$



MAX-FLOW MIN-CUT THEOREM

For any flow f and any cut $(S, \bar{S})$,

$$\text{value}(f) \leq \text{capacity}(S, \bar{S})$$

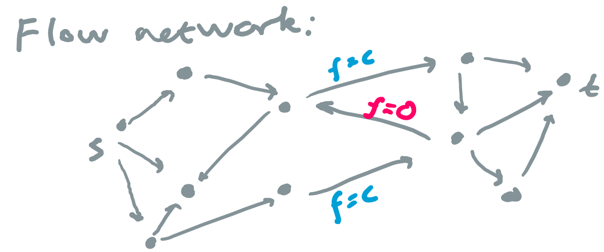
CORRECTNESS THEOREM

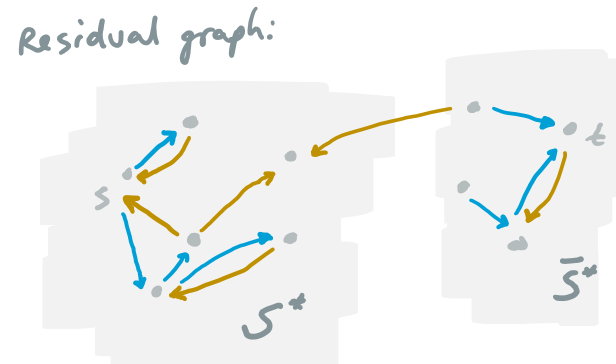
Suppose Ford-Fulkerson terminates, producing a flow f^*. Then f^* is a maximum flow.

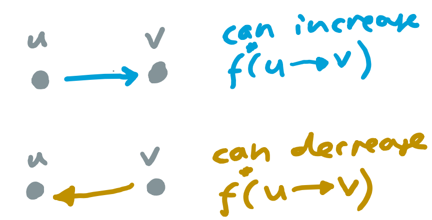
```
def ford_fulkerson(g, s, t):
    # Let f be a flow, initially empty
    for u → v in g.edges:
        f(u → v) = 0

    # Define a helper function for finding an augmenting path
    def find_augmenting_path():
        # Define the residual graph h on the same vertices as g
        for u → v in g.edges:
            if f(u → v) < c(u → v): give h an edge u → v labelled “inc u → v”
            if f(u → v) > 0: give h an edge v → u labelled “dec u → v”
        if h has a path from s to t:
            return some such path, together with the labels of its edges
        else:
            # Let S be the set of vertices the bandits can reach (used in the proof)
            return None

    # Repeatedly find an augmenting path and add flow to it
    while True:
        p = find_augmenting_path()
        if p is None:
            break
        else:
            compute δ, the amount of flow to apply along p, and apply it
            # Assert: δ > 0
            # Assert: f is still a valid flow
```



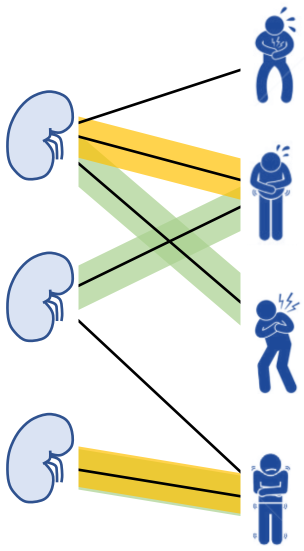




1. Let $S^* = \{\text{vertices reachable from } s\}$ in the residual graph, at termination.
2. The algorithm terminated, so $t \notin S^*$, so $(S^*, \bar{S}^*)$ is a cut.
3. The residual graph has no edges from S^* to $\bar{S}^*$, hence
    - on edges $S^* \rightarrow \bar{S}^*$ in the flow network, flow=capacity
    - on edges $S^* \leftarrow \bar{S}^*$ in the flow network, flow=0
4. From the inequalities in the max-flow min-cut theorem, $\text{value}(f^*) = \text{capacity}(S^*, \bar{S}^*)$; hence f^* is a maximum flow.

SECTION 6.4

Matchings

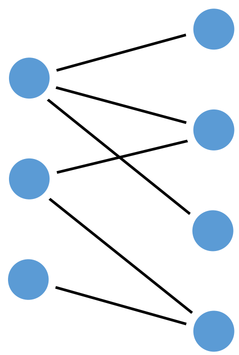


DEFINITIONS

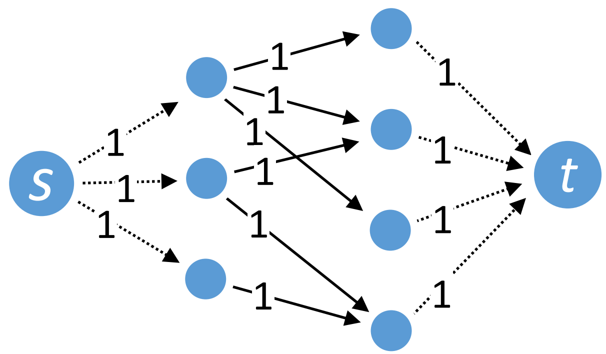
- A **bipartite graph** is an undirected graph in which the vertices are split into two sets, and all edges go between these sets
- A **matching** in a bipartite graph is a selection of edges, such that no vertex is connected to more than one of the edges
- The **size** of a matching is the number of edges it includes
- A **maximum matching** is one with the largest possible size

PROBLEM STATEMENT

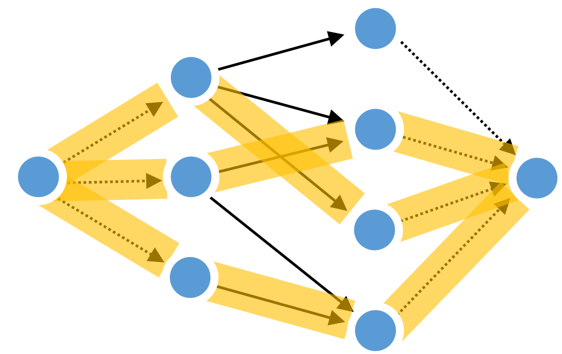
Given a bipartite graph, find a maximum matching



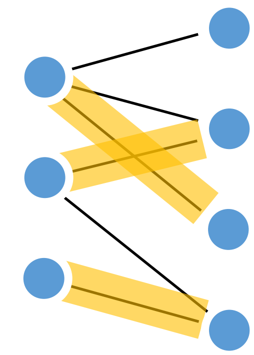
0. Given a bipartite graph
...



1. Build a helper graph:
- add source s and sink t
- add edges from s and to t

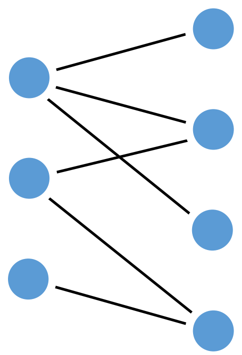


2. Solve max-flow on the helper graph, to find a maximum flow f^*

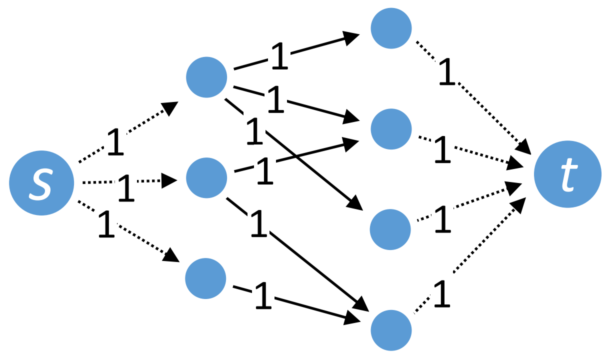


3. Interpret the flow f^* as a matching

What's the bug in my thinking?

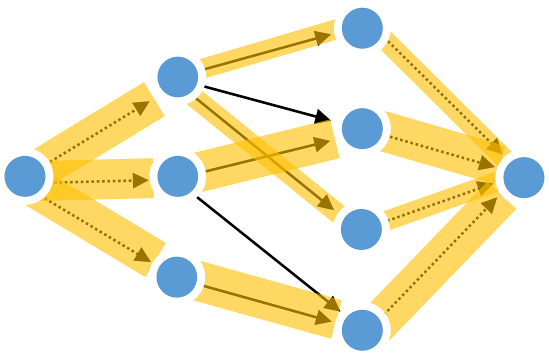


0. Given a bipartite graph
...



1. Build a helper graph:

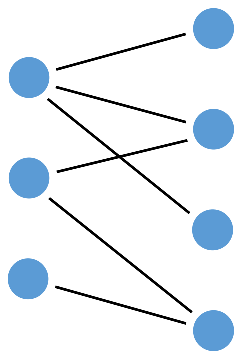
- add source s and sink t
- add edges from s and to t



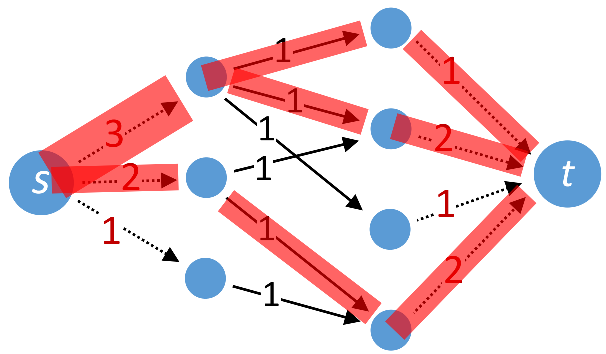
2. Solve max-flow on the helper graph, to find a maximum flow f^*

wtf ?!
This isn't the sort of flow I expected!

3. Interpret the flow f^* as a matching



0. Given a bipartite graph
...

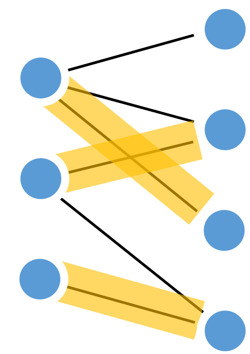


1. Build a helper graph:
- add source s and sink t
- add edges from s and to t

I'll set up a flow problem where the goal is to pick edges to **discard**.

Hold on! The max-flow solution actually leads to a **worse** matching.

2. Solve max-flow on the helper graph, to find a maximum flow f^*



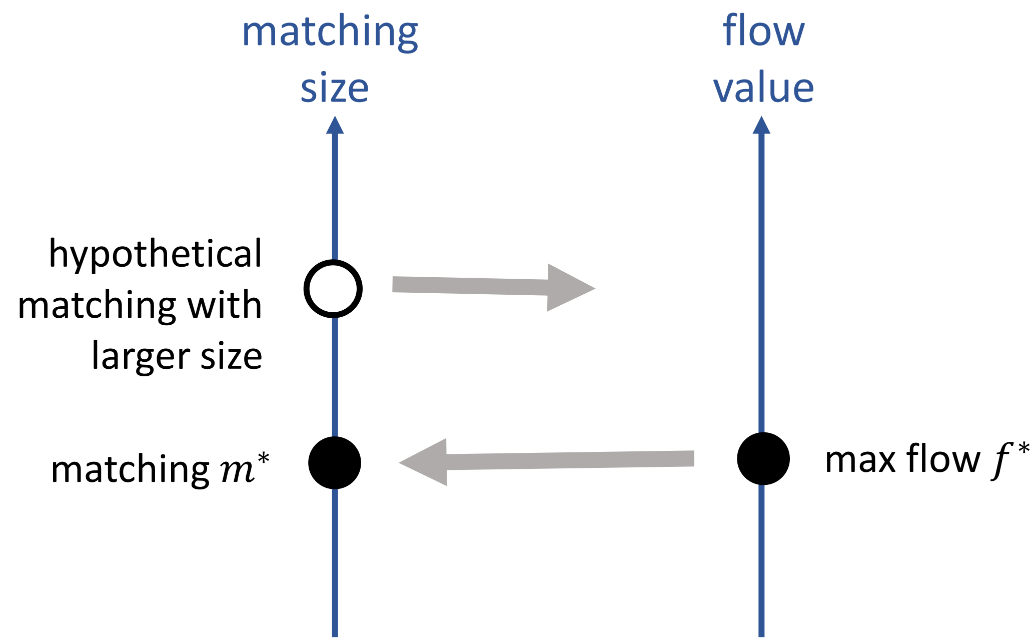
3. Interpret the flow f^* as a matching

THE TRANSLATION STRATEGY

CLAIM1: We can find a max flow f^* that can be translated into a matching, call it m^*

CLAIM2: If there were a larger-size matching m' then it would translate to a larger-value flow f'

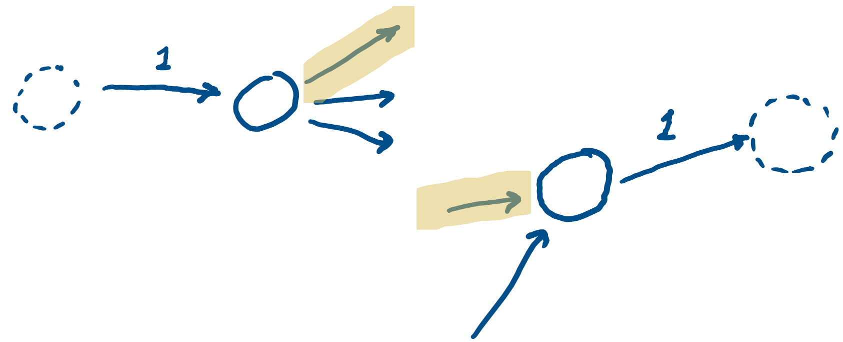
But there cannot be such a f', because f^* is a maximum flow. Therefore there is no such m', thus m^* is a maximum matching.



THE TRANSLATION STRATEGY

CLAIM1: We can find a max flow f^* that can be translated into a matching, call it m^*

Ford-Fulkerson will produce an integer flow, since all capacities are integer. Indeed, the flow on each edge must be either 0 or 1.



The capacity constraints tell us that, when we translate f^* into an edge selection, it meets the definition of "matching".

CLAIM2: If there were a larger-size matching m' then it would translate to a larger-value flow f'

But there cannot be such a f', because f^* is a maximum flow. Therefore there is no such m', thus m^* is a maximum matching.

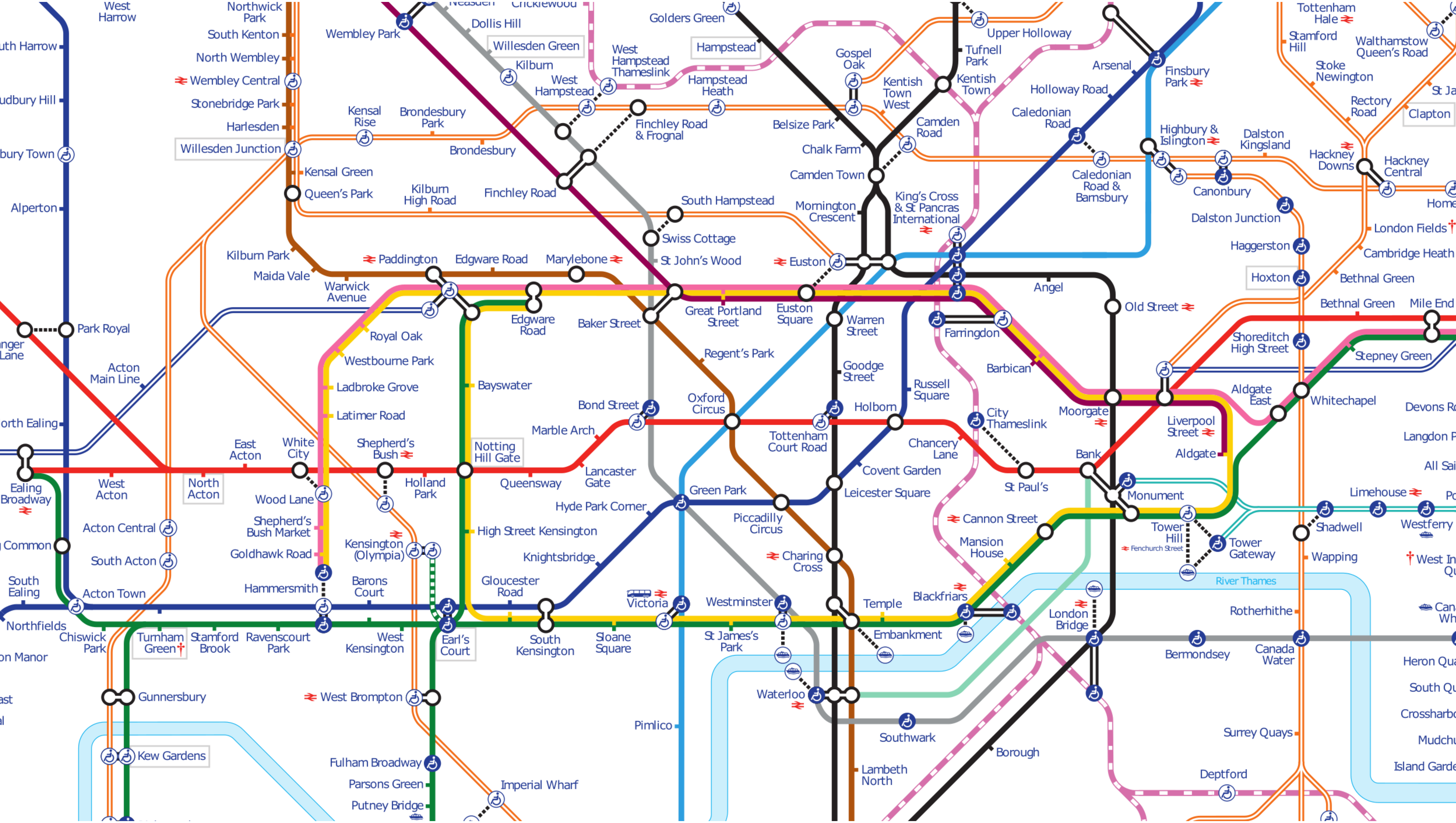
When we did the translation $f^* \to m^*$,

$$\text{value}(f^*) = \text{size}(m^*)$$

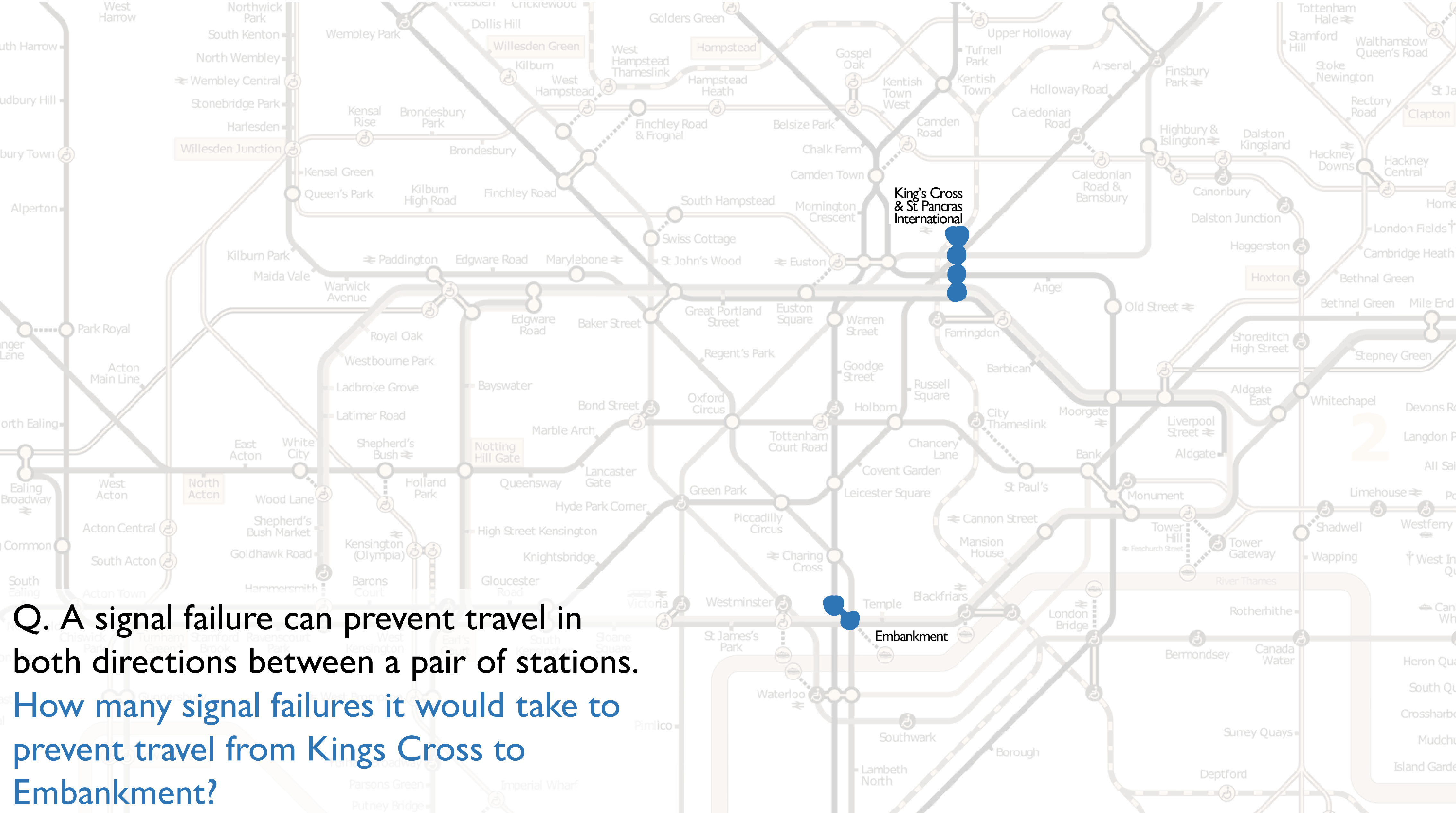
When we translate any matching to a flow, in the obvious way,

$$\text{value(flow)=size(matching)}$$

So if we had a larger-size matching m' it would translate to a larger-value flow f'.



West Harrow
Northwick Park
South Kenton
North Wembley
Wembley Central
Stonebridge Park
Harlesden
Willesden Junction
Kensal Green
Queen's Park
Kilburn Park
Maida Vale
Warwick Avenue
Paddington
Edgware Road
Marylebone
Wembley Park
Dollis Hill
Willesden Green
Kilburn
West Hampstead
Kensal Rise
Brondesbury Park
Brondesbury
Kilburn High Road
Finchley Road
West Hampstead Thameslink
Finchley Road & Frognal
Golders Green
Hampstead
Hampstead Heath
South Hampstead
Swiss Cottage
St John's Wood
Belsize Park
Chalk Farm
Camden Town
Gospel Oak
Kentish Town West
Kentish Town
Camden Road
Tufnell Park
Upper Holloway
Mornington Crescent
King's Cross & St Pancras International
Euston
Euston Square
Great Portland Street
Baker Street
Regent's Park
Edgware Road
Warren Street
Goodge Street
Russell Square
Holborn
Tottenham Court Road
Oxford Circus
Bond Street
Marble Arch
Lancaster Gate
Queensway
Notting Hill Gate
Holland Park
Shepherd's Bush
White City
East Acton
North Acton
West Acton
Ealing Broadway
Park Royal
Acton Main Line
Alperton
Wood Lane
Shepherd's Bush Market
Goldhawk Road
Hammersmith
Royal Oak
Westbourne Park
Ladbroke Grove
Latimer Road
Bayswater
Kensington (Olympia)
Barons Court
West Kensington
Earl's Court
Gloucester Road
South Kensington
Sloane Square
Victoria
Knightsbridge
Hyde Park Corner
High Street Kensington
Green Park
Piccadilly Circus
Leicester Square
Covent Garden
Charing Cross
Westminster
St James's Park
Embankment
Temple
Blackfriars
Waterloo
Southwark
Lambeth North
Pimlico
Acton Central
South Acton
Acton Town
Chiswick Park
Turnham Green
Stamford Brook
Ravenscourt Park
Northfields
South Ealing
Gunnersbury
Kew Gardens
West Brompton
Fulham Broadway
Parsons Green
Putney Bridge
Imperial Wharf
Chancery Lane
City Thameslink
St Paul's
Farringdon
Barbican
Moorgate
Bank
Monument
Cannon Street
Mansion House
London Bridge
Borough
Tower Hill
Fenchurch Street
Tower Gateway
Angel
Old Street
Liverpool Street
Aldgate
Aldgate East
Shoreditch High Street
Whitechapel
Stepney Green
Bethnal Green
Mile End
Hoxton
Haggerston
Dalston Junction
Canonbury
Dalston Kingsland
Highbury & Islington
Caledonian Road & Barnsbury
Caledonian Road
Holloway Road
Arsenal
Finsbury Park
Tottenham Hale
Stamford Hill
Stoke Newington
Walthamstow Queen's Road
Rectory Road
Clapton
Hackney Downs
Hackney Central
London Fields
Cambridge Heath
Limehouse
Shadwell
Westferry
Wapping
Rotherhithe
Canada Water
Bermondsey
Surrey Quays
Deptford
River Thames



Q. A signal failure can prevent travel in both directions between a pair of stations.

How many signal failures it would take to prevent travel from Kings Cross to Embankment?