All-Pairs Shortest Path

<u>Input</u>: ~~dir~~ Graph $G=(V,E)$ where each edge (u,v) has weight $wt(u,v)>0$.

<u>Output</u>: For all pairs of vertices a and b, the length of the shortest path from a to b ($\delta(a,b)$).

<u>Option 1</u>: Run Dijkstra for each vertex.

This takes $|V| \times O((|V|+|E|)\log|V|) = O(|V|^2 \log|V| + |V||E|\log|V|)$

If $|E| = O(|V|)$, this is just $O(|V|^2 \log|V|)$.

$\Rightarrow$ Good option if G is <u>sparse</u>.

<u>Option 2</u>: Dynamic Programming (Floyd-Warshall)

① <u>Structure of the optimal solution</u>.

Consider the shortest path from a to b.
Suppose it has at least one interior vertex.
Let k be the interior vertex with highest index.

shortest paths with interior vertices $\leq k-1$.

② <u>Recurrence</u>

Let $dist(a,b,k)$ = length of the shortest path from a to b with interior vertices $\leq k$.

We want to compute $dist(a,b,|V|) = \delta(a,b)$ for all $a,b \in V$.

$$dist(a,b,k) = \begin{cases} 0 & \text{if } a=b \\ wt(a,b) & \text{if } a \neq b,\ k=0,\ (a,b) \in E \\ \infty & \text{if } a \neq b,\ k=0,\ (a,b) \notin E \\ \min(dist(a,b,k-1),\ dist(a,k,k-1)+dist(k,b,k-1)) & \text{o.w.} \end{cases}$$

③ Solve bottom-up

Algorithm Floyd-Warshall (G):

```
for each vertex a do
  for each vertex b do
    if a = b then dist[a,b,0] ← 0
    else dist[a,b,0] ← ∞
for each edge (a,b) do
  dist[a,b,0] ← wt(a,b)
for k ← 1 to |V| do
  for each vertex a do
    for each vertex b do
      dist[a,b,k] ← min(dist[a,b,k-1],
                        dist[a,k,k-1] + dist[k,b,k-1])
```

Correct? Yes, by ① and ②.

Terminates? Yes.

Efficient? $O(|V|^3)$

$\Rightarrow$ Good option if G is dense.

Minimum Spanning Tree

__Input:__ Connected undirected graph G where each edge (u,v) has weight $wt(u,v)$.

__Output:__ Graph G' such that:

- $V' = V$ (same vertices)
- G' is connected
- weight(G') = sum of the weights of the edges in G'

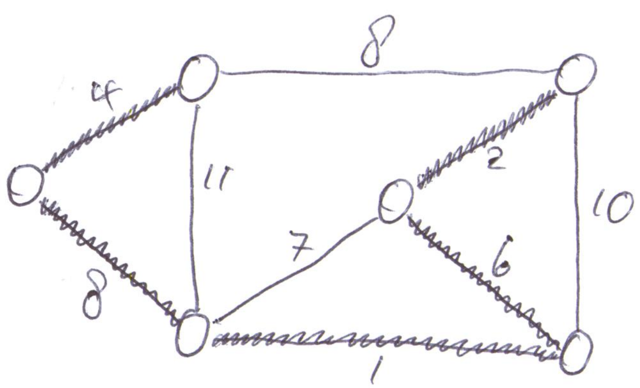
is ~~minim~~ __minimum__

__Claim:__ G' must be a __tree__ (connected and acyclic)

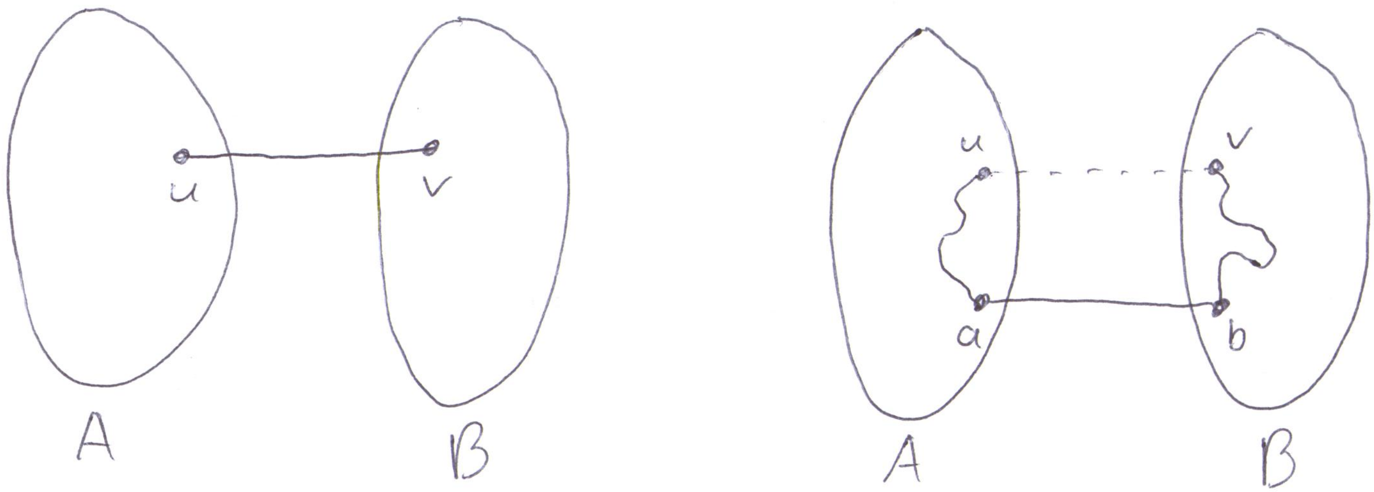
__Proof sketch:__ If not, it has a cycle and we can remove an edge, giving lower weight.

G' is a __minimum spanning tree__ of G.

__Example:__



Lemma: Partition V into A and B. Let (u,v) be the shortest edge connecting A and B. Then there is an MST of G that contains (u,v).



Proof: Let T be an MST of G.

If (u,v) is an edge of T, we are done.

Suppose that it is not.

Since T is connected, there is a path from u to v that uses an edge (a,b) with $a \in A$ and $b \in B$.

Let $T' = T$ minus (a,b), plus (u,v). T' is still a tree.

$$\text{weight}(T') = \text{weight}(T) - wt(a,b) + \underbrace{wt(u,v)}_{\leq wt(a,b)}$$

$$\leq \text{weight}(T)$$

Thus, T' is an MST that contains (u,v) $\square$