- Example 1

an edge \( e \) is in an MST if and only if \( e \) is the min weight edge in some cut.

\( e \) max weight, also min weight edge in some cut.

- Example 2

1. backward edge

backward edge if and only if graph has a cycle.

2. crossing edge

forward edge \( \checkmark \)

crossing edge \( \times \)
③ forward edge

Claim: only one of ① ② ③ can be in DFS tree.
the other two edges will both be forward edges.

④ original problem

colorful path

idea: try to remember color of last edge.

duplicate each vertex to \( k \) vertices

\[
\begin{align*}
U & \quad u_1, 0 \\
u_2 & \quad 0, u \\
& \vdots \\
U_k & \quad 0, u_i
\end{align*}
\]

reaching \( u_i \): there is a colorful path from \( s \) to \( u \) in original graph.
reaching \( u \): there is a colorful path from \( s \) to \( u \) in original graph and the last edge of the path is of color \( c \).

original graph: \( (U, V) \) of color \( c \)

\( (K-1) \) edges in the new graph

do DFS/BFS in the new graph

running time: constructing new graph \( O(mk) \)
running DFS/BFS \( O(mk) \)

define variables
for every edge \((i,j)\) let

\[ x_{ij} = \begin{cases} 1 & \text{if } (i,j) \text{ in path } 1 \\ 0 & \text{not in } 1 \end{cases} \]

\[ y_{ij} = \begin{cases} 1 & \text{if } (i,j) \text{ in path } 2 \\ 0 & \text{not in } 2 \end{cases} \]
define variables

for every edge \((i,j)\) let \(X_{i,j} = \begin{cases} 1 & \text{if } (i,j) \text{ in path 1} \\ 0 & \text{if } (i,j) \text{ not in path 1} \end{cases}\)

\[ Y_{i,j} = \begin{cases} 1 & \text{if } (i,j) \text{ in path 2} \\ 0 & \text{if } (i,j) \text{ not in path 2} \end{cases}\]

write constraints

for every edge \((i,j)\)

\[ 0 \leq X_{i,j} \leq 1 \]

\[ 0 \leq Y_{i,j} \leq 1 \]

\(X_{i,j}\) should form a path from \(s\) to \(t\)

for starting vertex \(s\), there is one outgoing edge

\[ \sum_{i, (i,i) \in E} X_{i,i} = 1 \]

for every intermediate vertex, \# incoming = \# outgoing

\[ \sum_{i, (i,j) \in E} X_{i,j} = \sum_{j, (i,j) \in E} X_{j,i} \]

# outgoing edges from \(s\) # incoming edges to \(s\)

for ending vertex \(t\), there is one incoming edge

\[ \sum_{i, (i,t) \in E} X_{i,t} = 1 \]

\# incoming edges to \(t\)

repeat for \(Y_{i,j}\)

two paths should not intersect

for every edge \((i,j)\)

\[ X_{i,j} + Y_{i,j} \leq 1 \]

objective function: sum of length of two paths
\[ \min \sum_{(i,j) \in E} w(i,j) \times (x_{ij} + y_{ij}) \]

Easier solution using max flow

\[ (z_{ij} = x_{ij} + y_{ij}) \]

\[ z_{ij} = \begin{cases} 1 & \text{if } (i,j) \text{ in one of the paths} \\ 0 & \text{otherwise} \end{cases} \]

\[ 0 \leq z_{ij} \leq 1 \]

Same flow conservation constraints,

\[ \sum_{i, (s,i) \in E} z_{si} = 2 \quad \sum_{i, (i,t) \in E} z_{it} = 2 \]

\[ \min \sum_{(i,j) \in E} w(i,j) \times z_{ij} \]