1.5em 0pt
CS218, Spring 2025, Due: 11:59pm, Fri. 06/05, 2025
Assignment #5 [1] Training Programming Due: 11:59pm, Fri. 05/30, 2025
Name: Lakhan Kumar Sunilkumar Student ID: 862481700

---

1 (1.5 pt) Union-Find

In the **union-by-height** strategy, we always attach the root of the *shallower tree* to the root of the *deeper tree*. Ties are broken arbitrarily (e.g., incrementing the height by one if heights are equal). Our goal is to prove that the height of any resulting tree after a sequence of unions is bounded by $O(\log n)$, where n is the total number of nodes.

So we know that any tree of height h has at least 2^h nodes. Since a tree cannot contain more than n nodes, this implies $2^h \leq n$, or equivalently, $h \leq \log_2 n$. We prove this formally via induction.

Base case: At the beginning, each node is in its own singleton tree of height 0 and size 1. This satisfies $1 = 2^0$.

Inductive Hypothesis: Assume that all trees of height up to h contain at least 2^h nodes.

Now, consider performing a union between two trees T_1 and T_2 with heights h_1 and h_2, respectively, and let $h = \max(h_1, h_2)$.

- **Case 1:** $h_1 \neq h_2$**.** Without loss of generality, assume $h_1 < h_2$. In this case, the shorter tree T_1 is attached under the taller tree T_2, and the new height remains $h = h_2$. By the inductive hypothesis,

$$|T_2| \geq 2^{h_2}, \quad |T_1| \geq 2^{h_1} \geq 1, \quad \Rightarrow \quad |T| = |T_1| + |T_2| \geq 2^{h_2} = 2^h.$$

- **Case 2:** $h_1 = h_2 = h$**.** We attach one tree under the other, and the resulting height increases by one to $h + 1$. By the inductive hypothesis:

$$|T_1| \geq 2^h, \quad |T_2| \geq 2^h \quad \Rightarrow \quad |T| = |T_1| + |T_2| \geq 2^{h+1}.$$

  Hence, the new tree of height $h + 1$ has at least 2^{h+1} nodes.

This completes the inductive proof: every tree of height h has size at least 2^h.

Final conclusion: Since the number of nodes in any tree cannot exceed n, the height of the tallest tree is bounded by:

$$2^h \leq n \quad \Rightarrow \quad h \leq \log_2 n \quad \Rightarrow \quad h = O(\log n).$$

Thus, union-by-height ensures that the depth of any node in the forest is at most $O(\log n)$, completing the proof.

[1] Some of the problems are adapted from existing problems from online sources. Thanks to the original authors.

2 (1.5 pt) Network Flow

1. (0.3pts) **One possible augmenting path is**

$$P \;=\; s \;\to\; 4 \;\to\; 8 \;\to\; t.$$

Along this path the residual capacities are

$$c(s,4)=6, \quad c(4,8)=3, \quad c(8,t)=5,$$

so the flow increment is

$$\Delta f \;=\; \min\{6,\,3,\,5\} \;=\; 3.$$

2. (0.6pts)

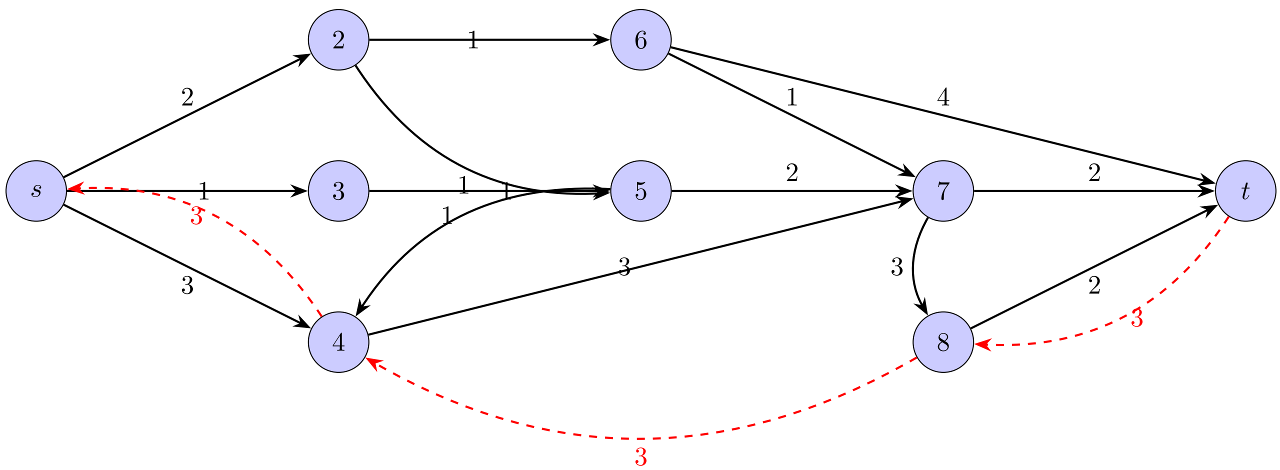


Figure 1: Residual network after augmenting along $s \to 4 \to 8 \to t$ by $\Delta f = 3$.

3. (0.6pts) **Maximum flow via Ford–Fulkerson**

Apply Ford–Fulkerson to the network shown above. We exhibit five augmenting paths and their increments:

Path	Edges	Δf	Bottleneck
P_1	$s \to 4 \to 8 \to t$	3	$\min\{3,\,3,\,2\} = 3$
P_2	$s \to 4 \to 7 \to t$	2	$\min\{1,\,1,\,2\} = 2$
P_3	$s \to 2 \to 6 \to t$	1	$\min\{2,\,1,\,4\} = 1$
P_4	$s \to 3 \to 5 \to 7 \to 8 \to t$	1	$\min\{1,\,1,\,2,\,3,\,2\} = 1$
P_5	$s \to 2 \to 5 \to 7 \to 8 \to t$	1	$\min\{1,\,1,\,1,\,1,\,2\} = 1$

After P_5, no augmenting path remains in the residual graph. Therefore the maximum flow is

$$\boxed{f_{\max} = 3 + 2 + 1 + 1 + 1 = 8.}$$

3 (1 pt) Bipartite Graph Matching

1. (0.5pts) **Augmenting path.**

   Start from the free vertex 3, follow an unmatched edge to b, then the matched edge back to 1, and finally an unmatched edge to the free vertex a:

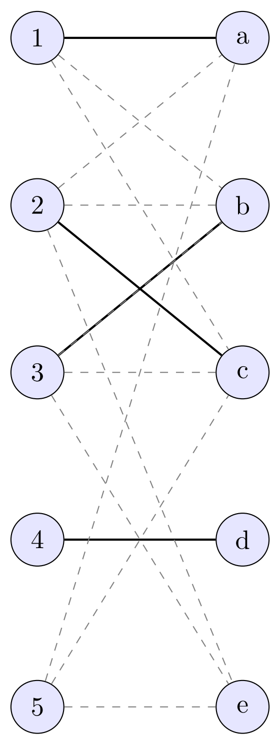
$$3 \xrightarrow{\text{unmatched}} b \xrightarrow{\text{matched}} 1 \xrightarrow{\text{unmatched}} a.$$

2. (0.5pts) **Updated matching.**

   Flip the status of edges along that path. The new matching becomes

$$\{1{-}a,\ 2{-}c,\ 3{-}b,\ 4{-}d\}.$$

   Graphically (solid = matched, dashed = other edges):



4 (1.5 pt) Strongly Connected Components

1. (0.3 pt) **Definition of SCC**

   A *strongly connected component* (SCC) of a directed graph $G = (V, E)$ is a maximal subset $C \subseteq V$ such that for every pair of vertices $u, v \in C$, there is a directed path from u to v and a directed path from v to u.

2. (0.4 pt) **SCC containing vertex X.**

   The strongly connected component that contains X is

   $$\{\, A,\ C,\ X \,\},$$

   since

   $$X \to C \to A \to X$$

   forms a directed cycle.

3. (0.6 pt) **BGSS partition around vertex X.**

   First compute two reachability sets in plain LaTeX (no custom macros):

   $$R_{\text{out}}(X) \;=\; \{\, v \in V \mid \text{there is a directed path } X \to v \}, \qquad R_{\text{in}}(X) \;=\; \{\, v \in V \mid \text{there is a directed path } v \to X \}.$$

   Then the four parts are

   $$\begin{aligned} S_1 &= R_{\text{out}}(X) \setminus R_{\text{in}}(X), \\ S_2 &= R_{\text{in}}(X) \setminus R_{\text{out}}(X), \\ S_3 &= R_{\text{in}}(X) \cap R_{\text{out}}(X), \\ S_4 &= V \ \setminus \ \big(R_{\text{in}}(X) \cup R_{\text{out}}(X)\big). \end{aligned}$$

   For the given graph one finds

   $$\begin{aligned} R_{\text{out}}(X) &= \{A, B, C, D, E, F, H, X\}, & R_{\text{in}}(X) &= \{I, J, A, B, C, D, E, F, H, X\}, \\ S_1 &= \varnothing, & S_2 &= \{I, J\}, \\ S_3 &= \{A, B, C, D, E, F, H, X\}, & S_4 &= \{G\}. \end{aligned}$$

4. (0.2 pt) There are 4 strongly connected components in the graph:

   $$\{A, B, C, D, E, F, H, X\}, \quad \{I\}, \quad \{J\}, \quad \{G\}.$$