Ford-Fulkerson

initialize \( f = 0 \), \( G_f = G \)

while there exists an augmenting path \( P \) in \( G_f \):
    compute \( \delta(P) = \min \) residual capacity on \( P \) \( \in O(n) \)
    update \( f \) to \( f + \delta(P) \cdot f^P \) \( \in O(n) \)
    recompute \( G_f \)
endwhile

output \( f \)

Correctness condition on termination: if we stop, we output a feasible flow with no augmenting path.

\([\text{Int.}]: f < f + \delta(P) \cdot f^P \) preserves feasibility \( \implies f \) is max flow.

\(O(m)\) time per iteration.

If capacities are integer valued, then

(a) \( f \) will always be an integer valued flow
(b) \( G_f \) will have integer capacities

\( \text{val}(f) \) increases by at least 1 per iteration, so the number of iterations bounded by \( \text{val}(f^*) \)
where \( f^* \) is a max flow.

Overall running time \( O(m \cdot \text{val}(f^*)) \).
Let $r = \frac{1}{2} (\sqrt{5} - 1)$ be the positive solution of $r + r^2 = 1$.

This iterates without end.

Algorithm

Ford-Fulkerson

Edmonds-Karp #1
augmenting path that maximizes $\delta(P)$

Edmonds-Karp #2
augmenting path with fewest edges

Running Time

\[ O(m \cdot \text{val}(f^*)) \]

\[ O(m \cdot \log(n) \cdot \log(n \cdot \text{val}(f^*))) \]

\[ O(m^2n) \]

pseudopolynomial (poly if numbers encoded in unary)

weakly polynomial (poly in input size)

strongly polynomial (polynomial with no dependence on # of digits, as long as arithmetic takes $O(d)$)
Dinitz

Push-Relabel

O(n^3)

Orlin's Algorithm

O(mn)

Chen, Kyng, Lie, Peng, Robertson, and Sachdeva (2022)

O(m(n + d) log val(f^*)) w.h.p.,

Fastest known strongly poly

Push-Relabel Algorithm

Maintains two objects: a preflow and a height function.

**Def.** \( f \) is a preflow if it satisfies

1. \( f(u, v) + f(v, u) = 0 \quad \forall u, v \)
2. \( \sum_{u \in V} f(u, v) \geq 0 \quad \forall v \neq s \) ("excess" of \( v \), net flow arriving into \( v \)),