SoC 2023: Competitive Programming

Week-7: Flows and Tries

Mentor: Virendra Kabra

Summer 2023

Contents

Reference: cp-algos, CLRS

1 Flows

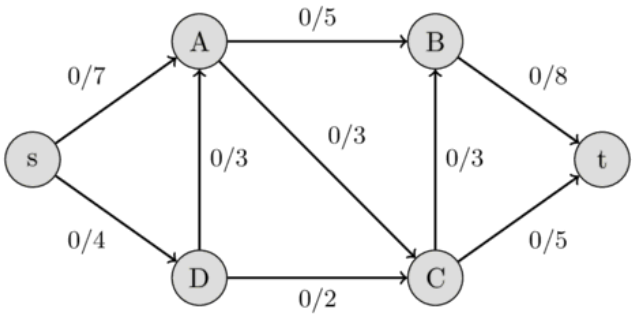
1.1 Terminology

- Network: Directed graph with a capacity function $c : E \to \mathbb{R}_0^+$. Special vertices source s and sink t. Analogous to a network of water pipes.
- Flow: Function f assigning flows to edges such that for all $e \in E$, $f(e) \leq c(e)$. Further, for all $u \in V \backslash \{s, t\}$, incoming and outgoing flows must be equal:

$$\sum_{(v,u)\in E} f((v,u)) = \sum_{(u,v)\in E} f((u,v))$$

- Can deduce

$$\sum_{(s,u)\in E} f((s,u)) = \sum_{(u,t)\in E} f((u,t))$$

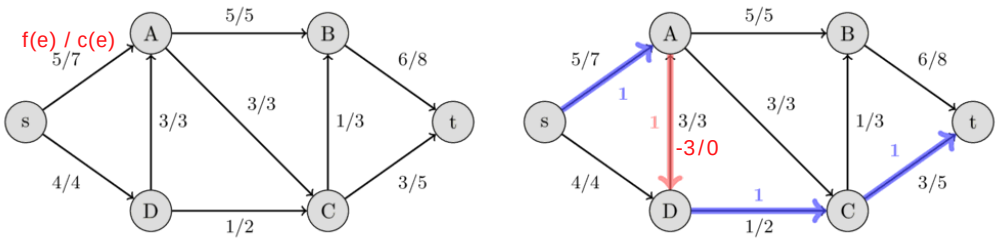


- Value of a flow: Sum of outgoing flows from source. Equivalent to sum of incoming flows to sink.
- Maximal flow: Flow with maximum value.

We need to find a maximal flow.
maximum

1.2 Ford-Fulkerson Method

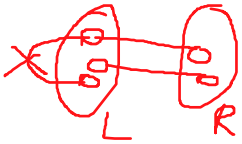
- Residual capacity of an edge: $r(e) = c(e) - f(e)$. By an earlier constraint, this is non-negative.
- Residual network: Add (directed) reverse edges, with zero capacity and $-f(e)$ flow. Note that reverse edges also have non-negative residual flow. These can be used to reverse an earlier assigned flow. For example, (A, D) has a residual capacity of 3. "Using" it is same as undoing (D, A) flow:



- Augmenting path: A simple path in the residual network, along edges with positive residual capacity.
- Pseudocode

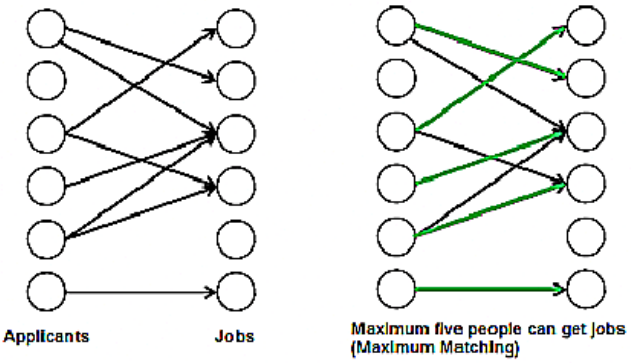
```
while there is an augmenting path P from s to t:
    c = capacity of P
    for e in P:
        r(e) -= c
        r(e_reverse) += c
max_flow = sum of outgoing flows from s
```

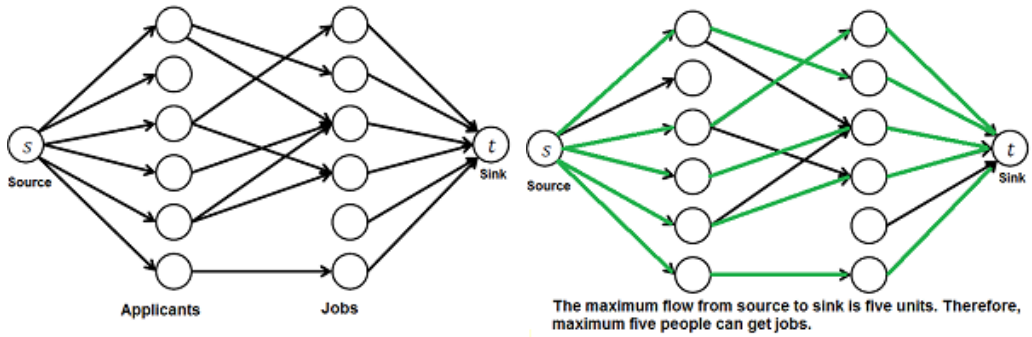
- Edmonds-Karp implementation: Use BFS to find augmenting paths.



1.3 Example: Maximum Bipartite Matching

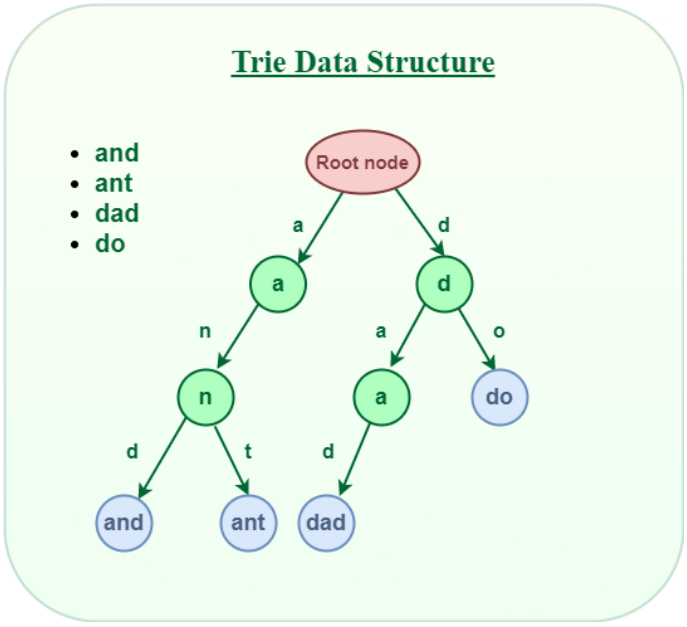
- X applicants and Y jobs. Each applicant x applies for some set of jobs Y_x. Each job can only admit one applicant, and each applicant can only get one job. Find an assignment of jobs that fills maximum number of vacancies.
- Matching of a graph: Set of edges such that on any vertex, at most one of these is incident.
- Maximum matching: Any matching with the largest number of edges.
- We represent this as a bipartite graph, and the problem reduces to find its maximum matching.



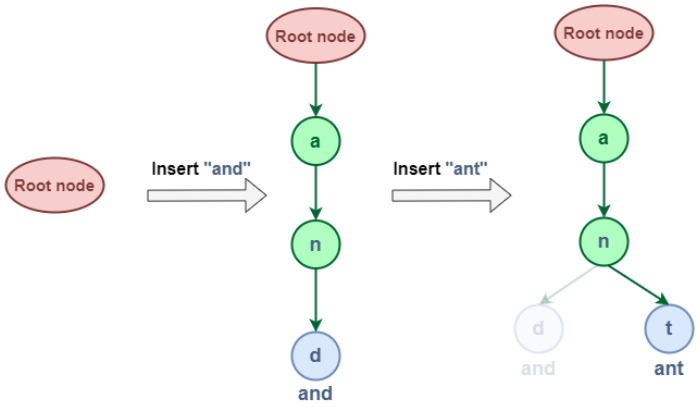


2 Tries

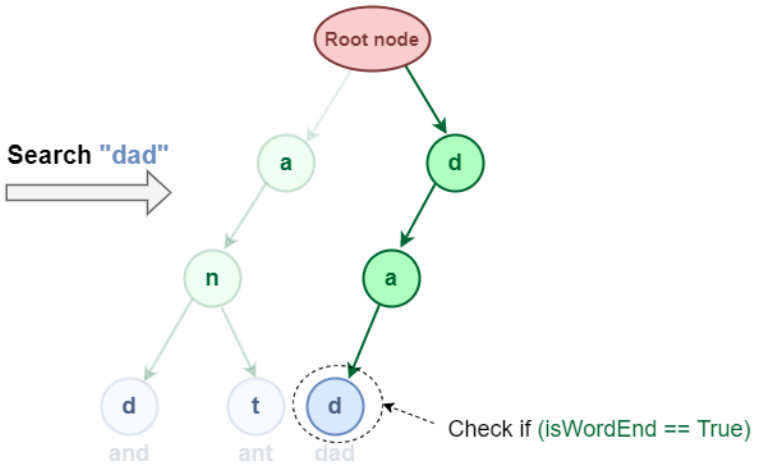
Reference: gfg.
To efficiently store and search strings.



Insertion



Search



3 Todos

Check sheet.