CSE 431/531: Analysis of Algorithms (Summer 2023)
All Pair Shortest Path

Chen Xu

July 27, 2023

All Pair Shortest Path

All Pair Shortest Path Problem

Input: Weighted Directed Graph $G = (V, E, W)$

Output: A matrix f s.t.
$f[i][j]$ is the length of the shortest path from i to j

DP approach

- Let $w(i,j) = \begin{cases} 0 & i = j \\ \text{weight of edge } (i,j) & i \neq j, (i,j) \in E \\ \infty & i \neq j, (i,j) \notin E \end{cases}$
- Let $f^k[i][j]$ be the subproblem of length of shortest path from i to j that only uses vertices $\{1, 2, 3, ..., k\}$ as intermediate vertices. We have the recursive structure:

$$f^k[i,j] = \begin{cases} w(i,j) & k = 0 \\ \min \begin{cases} f^{k-1}[i,j] \\ f^{k-1}[i,k] + f^{k-1}[k,j] \end{cases} & k = 1, 2, \cdots, n \end{cases}$$

Floyd-Warshall algorithm

- Similar to Bellman-Ford algorithm, we just need to keep one copy of the matrix. The optimized version is as follows:

Floyd-Warshall APSP

```
Initialize f ← w
for k ← 1 to n do
    for i ← 1 to n do
        for j ← 1 to n do
            if f[i,k] + f[k,j] < f[i,j] then
                f[i,j] ← f[i,k] + f[k,j]
```

- Running time is $O(n^3)$.

Summary of shortest path problem

Algorithm	**Weights**	**Single Source?**	**Running Time**
Dijkstra	$\mathbb{R}_{\geq 0}$	SS	$O(n \log n + m)$
Bellman-Ford	$\mathbb{R}$	SS	$O(nm)$
Floyd-Warshall	$\mathbb{R}$	AP	$O(n^3)$

Table: Summary of Shortest Path Algorithms