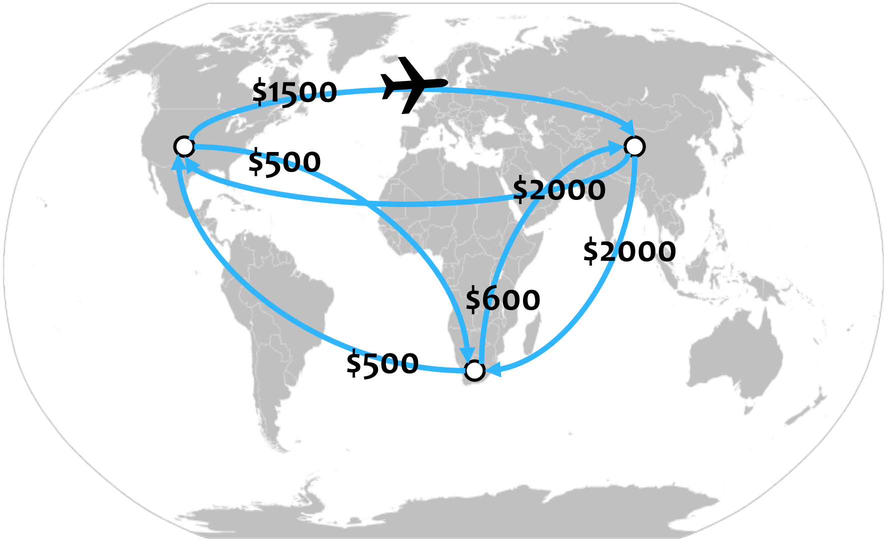
The Flights Problem

$c(i, j)$ might be different from $c(j, i)$

* Given n airports and *possibly asymmetric* costs $c(i, j)$ to directly fly from airport i to j
* **Goal:** for *every* pair of airport i, j, find the *minimum* cost, $d(i, j)$, to fly from airport i to airport j, *when layovers are allowed*



Recurrence for the Flights Problem?

* Given n airports and possibly asymmetric costs $c(i, j)$ to directly fly from airport i to j
* **Definition:** Let $d^k(i, j)$ be the minimum cost to fly from i to j (the cost of a *cheapest* path from i to j) that allows layovers only in airports $1, \ldots, k$.

Floyd–Warshall Algorithm

Recurrence:

$$d^k(i,j) = \begin{cases} c(i,j) & \text{if } k = 0 \\ \min\{d^{k-1}(i,j), d^{k-1}(i,k) + d^{k-1}(k,j)\} & \text{if } 0 < k \leq n \end{cases}$$

Bottom-up Algorithm:

Floyd-Warshall($C[1..n][1..n]$): // $C[i][j] = c(i,j)$

allocate $D[1..n][1..n][0..n]$ // $D[i][j][k] = d^k(i,j)$

for all i,j: $D[i][j][0] \leftarrow C[i][j]$

for $k = 1..n$ and all i,j:

$$D[i][j][k] \leftarrow \min\begin{Bmatrix} D[i][k][k-1] + D[k][j][k-1], \\ D[i][j][k-1] \end{Bmatrix}$$

Runtime: $O(n^3)$

ECE CSE

Hard to memorize..

Incorrect implementations of the Floyd–Warshall algorithm give correct solutions after three repeats

Ikumi Hide
The University of Tokyo
ihide@es.a.u-tokyo.ac.jp

Soh Kumabe
The University of Tokyo
sohkuma0213@gmail.com

Takanori Maehara
RIKEN Center for Advanced Intelligence Project
takanori.maehara@riken.jp

Abstract

The Floyd–Warshall algorithm is a well-known algorithm for the all-pairs shortest path problem that is simply implemented by triply nested loops. In this study, we show that the incorrect implementations of the Floyd–Warshall algorithm that misorder the triply nested loops give correct solutions if these are repeated three times.

Keywords: graph algorithm; algorithm implementation; common mistake

1. Introduction

The *Floyd–Warshall algorithm* is a well-known algorithm for the all-pairs shortest path problem [1, 2]. Let $G = (V, E)$ be a complete directed graph, where $V = \{1, \dots, n\}$ is the set of vertices, and let $w \colon V \times V \to \mathbb{R} \cup \{\infty\}$ be the length of the edges. We assume that G has no negative cycles. The Floyd–Warshall algorithm maintains array $\mathtt{d}$ of size $n \times n$ initialized by $\mathtt{d}[i, j] \leftarrow w(i, j)$ for all $i, j \in V$, and performs triply nested loops as shown in Algorithm 1. Then

Connection to Matrix Multiplication?

* Given n airports and possibly asymmetric costs $c(i, j)$ to directly fly from airport i to j
* **Definition:** Let $d^k(i, j)$ be the minimum cost to fly from i to j ***in at most k flights.***

Connection to Matrix Multiplication?

* When $k = 1$,

$$d^1(i,j) = c(i,j)$$

* When $k > 1$,

$$d^k(i,j) = \min_{\ell \in [n]} \{d^{k-1}(i,\ell) + c(\ell,j)\}$$

* Let us change "min" to "$\sum$", and "+" to "$\cdot$" from above

$$d^k(i,j) = \sum_{\ell \in [n]} d^{k-1}(i,\ell) \cdot c(\ell,j)$$

$$d^k = d^{k-1} \cdot c \;\; \textit{(as n by n matrices)}$$

* d^k is exactly c^k (c multiplied k times)!