Weighted Graphs

Việt Tùng Hoang

The slides are loosely based on those of Dr Kevin Wayne, Princeton University and Prof. Mary Wootters, Stanford University
1. Dijkstra’s algorithm

2. Minimum spanning Tree
Shortest Paths from CS Department to Airport

- **17 min**
- **7.0 miles**

- **17 min**
- **6.5 miles**
Shortest Path Problem

Find the shortest path from A to H
Shortest Path Problem

Find the shortest path from A to H

Run BFS: A → E → H
Shortest Path Problem

Find the shortest path from A to H

Run BFS: A → E → H

Doesn’t work if the edges are labeled by distance
Shortest Path Problem

Weighted graph

For now, weights are non-negative

$w(u, v)$: weight of edge between $u$ and $v$
Shortest Path Problem

Weighted graph

For now, weights are non-negative

If I care about the weights, I should go $A \rightarrow B \rightarrow G \rightarrow H$

$w(u, v)$ : weight of edge between $u$ and $v$
Single-Source Shortest Path Problem

Find the shortest paths from A to every other node

<table>
<thead>
<tr>
<th>Destination</th>
<th>Cost</th>
<th>How to get there</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>1</td>
<td>A → B</td>
</tr>
<tr>
<td>C</td>
<td>17</td>
<td>A → C</td>
</tr>
<tr>
<td>D</td>
<td>15</td>
<td>A → D</td>
</tr>
<tr>
<td>E</td>
<td>23</td>
<td>A → B → E</td>
</tr>
</tbody>
</table>

(Not necessarily stored as a table – how this information is represented will depend on the application)
An Application: Network Routing
Dijkstra’s Algorithm

Solve the single-source shortest path problem in $O(m \log(n))$ time

**Restriction:** weights must be non-negative
Exercise: Graphs With Node Weights

Here nodes (instead of edges) are labeled with weights.
Exercise: Graphs With Node Weights

This path has cost 33
How to find shortest paths from A to every other nodes?
Transformation to Edge Weights

A — 9 — B

↓

A — 5 — B
Transformation to Edge Weights

A → B: 9
A → D: 7
B → C: 5
B → D: 12
C → A: 9
C → D: 12
D → A: 5
D → B: 5
D → C: 12
Transformation to Edge Weights

Shortest paths from $s$ to $t$ remain the same, but the cost decreases by weight(s)
Exercise: Nearest Airports

Goal: For each city, find its nearest airport
Exercise: Nearest Airports

**Goal:** For each city, find its nearest airport

![Graph with cities and distances](image)

- **City with airport**
- **City without airport**

The nearest airport of **X** is **Y**
Merging Airports

Find shortest paths from $s$ to every node

City with airport

City without airport
Shortest Paths

- dist(s, v)
- predecessor of X is Y
Nearest Airports

Cost: $O(m \log(n))$

Using Dijkstra
Exercise: Shortest Path With Edge Skipping

Goal: Minimize the shortest path from source to destination if we have the ability to set the weight of one edge to 0
Exercise: Shortest Path With Edge Skipping

If we pick (C, E) then the cost of the shortest path is now 3

Goal: Minimize the shortest path from source to destination if we have the ability to set the weight of one edge to 0
Exercise: Shortest Path With Edge Skipping

If we instead pick (B, E) then the cost of the shortest path is now 1

Goal: Minimize the shortest path from source to destination if we have the ability to set the weight of one edge to 0
Copying Graph
Copying Graph

```
A -- 1 -- 4 -- C
  |    |    |    |
  v    v    v    v
B 10    D 4    E
```

```
A' -- 1 -- 4 -- C'
  |    |    |    |
  v    v    v    v
B' 10    D' 4    E'
```
Copying Graph

Find shortest path from $A$ to $E'$
Shortest Paths

predecessor of $X$ is $Y$

$\text{dist}(A, v)$
Shortest Path With Edge Skipping

Cost: $O(m \log(n))$

Using Dijkstra
Agenda

1. Dijkstra’s algorithm

2. Minimum spanning Tree

3. Prim

4. Kruskal

5. Clustering
Connecting Freckles
There must be a sequence of connecting lines from any freckle to another
Minimum Spanning Tree
Say we have a weighted, undirected graph

A spanning tree is a tree that connects all nodes.

Recall: a tree is a connected graph without cycle
Minimum Spanning Tree

Say we have a weighted, undirected graph.

A spanning tree is a tree that connects all nodes.

The cost of a spanning tree is the sum of the weights on the edges.

This is a spanning tree. It has cost 66.
Minimum Spanning Tree

Say we have a weighted, undirected graph

A spanning tree is a tree that connects all nodes.

This is also a spanning tree
It has cost 37
Minimum Spanning Tree
Say we have a weighted, undirected graph

A minimum spanning tree is a spanning tree of minimal cost.
Minimum Spanning Tree

Say we have a weighted, undirected graph

A minimum spanning tree is a spanning tree of minimal cost.

This is a minimum spanning tree

It has cost 37
Why MST?

- Network design:
  Connect cities with roads/electricity/phones

- Clustering

- Useful tool for other algorithms

MST algorithms: Prim, Kruskal

$O(m \log(n))$ time
Exercise: Maximum Spanning Tree

Question: Find the maximum spanning tree
Exercise: Maximum Spanning Tree

This is a maximum spanning tree

Question: Find the maximum spanning tree
Updating Weights

Negate each weight and then add a big constant (11)
From MST to Maximum Spanning Tree

This is an MST
From MST to Maximum Spanning Tree

This is an MST

This is a Maximum Spanning Tree
Exercise: Minimum Bottleneck Spanning Tree

Bottleneck Spanning Tree:
Cost is measured as the weight of the **heaviest** edge of the tree
Exercise: Minimum Bottleneck Spanning Tree

This has cost 14
Exercise: Minimum Bottleneck Spanning Tree

Question: Find a Minimum Bottleneck Spanning Tree
Given a value $V$, determine if the graph has a bottleneck spanning tree of cost at most $V$
Delete Edges Whose Weights Are Bigger Than $V$

Graph is disconnected $\Rightarrow$ Answer: No

Cost: $O(m + n)$ $\leftarrow$ Using BFS/DFS
Given a value $V$, determine if the graph has a bottleneck spanning tree of cost at most $V$
Delete Edges Whose Weights Are Bigger Than $V$  

$V = 9$

Graph is connected  

Answer: Yes
Finding A Bottleneck Spanning Tree

Any spanning tree of the resulting graph is good

Cost: $O(m + n)$ ← Using BFS/DFS
Binary Search for The Optimal $V$

Sorted weight list: [1, 2, 4, 6, 7, 8, 9, 10, 11, 14]

Is there’s a bottleneck spanning tree with $V = 7$? No
Sorted weight list: [1, 2, 4, 6, 7, 8, 9, 10, 11, 14]

Is there’s a bottleneck spanning tree with $V = 10$?
Sorted weight list: [1, 2, 4, 6, 7, 8, 9, 10, 11, 14]

Is there’s a bottleneck spanning tree with $V = 10$? Yes
Sorted weight list: [1, 2, 4, 6, 7, 8, 9, 10, 11, 14]

Is there’s a bottleneck spanning tree with $V = 9$?
Binary Search for The Optimal $V$

Sorted weight list: $[1, 2, 4, 6, 7, 8, 9, 10, 11, 14]$

Is there’s a bottleneck spanning tree with $V = 9$? Yes
Sorted weight list: [1, 2, 4, 6, 7, 8, 9, 10, 11, 14]

Is there’s a bottleneck spanning tree with \( V = 8 \)?
Sorted weight list: [1, 2, 4, 6, 7, 8, 9, 10, 11, 14]

Is there's a bottleneck spanning tree with $V = 8$?  No
Binary Search for The Optimal $V$

Sorted weight list: [1, 2, 4, 6, 7, 8, 9, 10, 11, 14]

$V = 9$
Binary Search for The Optimal \( V \)

Find a bottleneck spanning tree for the optimal \( V \)

This is the minimum bottleneck spanning tree

Cost: \( O(m \log(n)) \)
Exercise: Water Supply

A house gets water by either having a well there, or by being connected to a house of well via pipes.
Exercise: Water Supply

This is a way to build pipes/wells, so that everyone has water.

It has cost 25
Here’s a better way to build pipes/wells, and everybody still gets water.

It has cost 15.
Exercise: Water Supply

Question: Build wells/pipes so that everybody has water, but minimize the cost.
Constructing MST
Optimal Water Supply

Cost: $O(m \log(n))$

Using Prim/Kruskal
A few *edges* must be included in our spanning tree

**Question**: Find a spanning tree subject that contains the required edges and has as small cost as possible
Finding MST
Getting Back Original MST

Cost: $O(m \log(n))$

Using Prim/Kruskal
Some Subtlety In Merging

What happens if we merge B and C?
Some Subtlety In Merging

There are now two edges between the nodes B/C and D.

Should keep just one of them, but which one?
Some Subtlety In Merging

There are now two edges between the nodes B/C and D.