Basic Graph Algorithms

Thursday, September 28, 2017  5:03 PM

Problem A: Arctic Network
Problem B: Island Hopping

• Minimum Spanning Tree

Minimum spanning tree - Wikipedia
https://en.wikipedia.org/wiki/Minimum_spanning_tree

• Prim’s Algorithm

Prim’s Algorithm

1. Given a network
2. Choose a vertex
3. Choose the shortest edge from this vertex.
4. Choose the nearest vertex not yet in the solution.
5. Choose the next nearest vertex not yet in the solution, when there is a choice choose either.
6. Repeat until you have a minimal spanning tree.

• Kruskal’s Algorithm

Kruskal’s Algorithm

1. Given a network
2. Choose the shortest edge (if there is more than one, choose any of the shortest)
3. Choose the next shortest edge and add it.
4. Choose the next shortest edge which wouldn’t create a cycle and add it.
5. Choose the next shortest edge which wouldn’t create a cycle and add it.
6. Repeat until you have a minimal spanning tree.

• Disjoint-set data structure / Union–find data structure
• Reference
  ○ [https://en.wikipedia.org/wiki/Minimum_spanning_tree](https://en.wikipedia.org/wiki/Minimum_spanning_tree)
Problem C: Nikola

- Solved by Ziyi Zhang 🎉
- | | | |from| | ... | |i| | ... | |to| | |
- Where from = i-(stepLength-1), to = i+stepLength
- mincost[i][j] = min(mincost[i+stepLength][stepLength], mincost[i-(stepLength-1)][stepLength])
- Time complexity: \( \Theta(n^2) \)

Problem D: Reachable Roads

- See "Disjoint-set data structure / Union–find data structure"

Problem E: Single source shortest path, negative weights

- Bellman–Ford algorithm

---

Bellman-Ford in 5 minutes — Step by step example

---

- Reference
  - https://en.wikipedia.org/wiki/Bellman%E2%80%93Ford_algorithm
  - https://www-m9.ma.tum.de/graph-algorithms/spp-bellman-ford/index_en.html (Highly recommended)

- Our Github repository: