Tree

Binglun Wang

University College London

11st Jan, 2023

Sliver algorithm

- Most frequent: Depth First Search

Sliver algorithm

- Most frequent: Depth First Search
- Second: Introduction to Prefix Sums, Custom Comparators, and Coordinate Compression

Sliver algorithm

- Most frequent: Depth First Search
- Second: Introduction to Prefix Sums, Custom Comparators, and Coordinate Compression
- Second (Cont.): Greedy Algorithms with Sorting, Binary Search, and Flood Fill

Sliver algorithm

- Most frequent: Depth First Search
- Second: Introduction to Prefix Sums, Custom Comparators, and Coordinate Compression
- Second (Cont.): Greedy Algorithms with Sorting, Binary Search, and Flood Fill
- Others: https://usaco.guide/silver/

Gold algorithm

- Most frequent: Introduction to DP

Gold algorithm

- Most frequent: Introduction to DP
- Second: Disjoint Set Union, Point Update Range Sum (Segment Tree, Binary Indexed Tree)

Gold algorithm

- Most frequent: Introduction to DP
- Second: Disjoint Set Union, Point Update Range Sum (Segment Tree, Binary Indexed Tree)
- Others: `https://usaco.guide/gold`

Gold algorithm

- Most frequent: Introduction to DP
- Second: Disjoint Set Union, Point Update Range Sum (Segment Tree, Binary Indexed Tree)
- Others: `https://usaco.guide/gold`
- What are the differences between algorithms of silver and gold?

How to train?

- Aim: Feb 24-27: Third Contest; Mar 24-27: US Open

How to train?

- Aim: Feb 24-27: Third Contest; Mar 24-27: US Open
- 1. Practices for the Algorithm list

How to train?

- Aim: Feb 24-27: Third Contest; Mar 24-27: US Open
- 1. Practices for the Algorithm list
- 2. Codeforces online contest: `https://codeforces.com/contests`

Recommended references

- USACO Guide `https://usaco.guide/dashboard`
- OIWiki `https://oi-wiki.org/`
- Matrix67 `http://www.matrix67.com/blog/`

What is Graph?

- $$G = (V(G), E(G)) \tag{1}$$

  Where V is the vertex set of G, E is the edge set of G

What is Graph?

- $$G = (V(G), E(G)) \tag{1}$$

  Where V is the vertex set of G, E is the edge set of G
- Examples: Image, 1-D Array, etc...

What is Graph?

- $$G = (V(G), E(G)) \tag{1}$$

  Where V is the vertex set of G, E is the edge set of G
- Examples: Image, 1-D Array, etc...
- graph concept: https://oi-wiki.org/graph/concept/

Undirected & directed edge|graph

- Examples of undirected graph: Wechat, Whatsapp, etc...

Undirected & directed edge|graph

- Examples of undirected graph: Wechat, Whatsapp, etc...
- Examples of directed graph: Youtube, Bilibili, Twitter, etc...

Degree

- In undirected graph: the number of edges of a node

Degree

- In undirected graph: the number of edges of a node
- In directed graph: in-degree, out-degree

Simple graph

- loop: An edge $e = (u, v)$, where $u = v$, we call the e is a loop

Simple graph

- loop: An edge $e = (u, v)$, where $u = v$, we call the e is a loop
- multiple edge: in E if it has two same edges $e_1 = e_2$.

Simple graph

- loop: An edge $e = (u, v)$, where $u = v$, we call the e is a loop
- multiple edge: in E if it has two same edges $e_1 = e_2$.
- simple graph: A graph W/O loop and multiple edges

Others

- cycle

Others

- cycle
- subgraph

Others

- cycle
- subgraph
- connected component

What is tree

- Question: for an undirected graph with N nodes, what is the minimum number of edges to obtain a connected graph?

What is tree

- Question: for an undirected graph with N nodes, what is the minimum number of edges to obtain a connected graph?
- The simplest structure of a connected undirected graph. (an undirected graph can be composed of multiple connected undirected graphs)

What is tree

- Question: for an undirected graph with N nodes, what is the minimum number of edges to obtain a connected graph?
- The simplest structure of a connected undirected graph. (an undirected graph can be composed of multiple connected undirected graphs)
- unrooted tree & rooted tree

What we want?

- 1. minimum upper bound subject to 2. Sorted

What we want?

- 1. minimum upper bound subject to 2. Sorted
- Direction 1: satisfy the subjection (Sorted) then find the minimum upper bound.

What we want?

- 1. minimum upper bound subject to 2. Sorted
- Direction 1: satisfy the subjection (Sorted) then find the minimum upper bound.
- Direction 2: We don't know the minimum, guess the minimum upper bound, the check is it satisfied the subjection.

Direction 2

- How to guess and how to check.

Direction 2

- How to guess and how to check.
- Time complexity?

Direction 2

- How to guess and how to check.
- Time complexity?
- $$O(AB) \tag{2}$$

  where A is the time complexity of 'guess', B is the time complexity of 'check'.

Guess

- Method?

Guess

- Method?
- Random?

Guess

- Method?
- Random?
- Iteration?

Guess

- Method?
- Random?
- Iteration?
- Monotonicity

Check

- Given a upper bound m

Check

- Given a upper bound m
- minimum cost function:

$$\sum_{i}^{n} f_m(a) - i \tag{3}$$

where a is the array of cow positions, f_m is the order of wormhole we use (width less than w)

Check

- Given a upper bound m
- minimum cost function:

$$\sum_{i}^{n} f_m(a) - i \tag{3}$$

where a is the array of cow positions, f_m is the order of wormhole we use (width less than w)

- A cow just needs to be able to swap to the corresponding position.

Check

- Given a upper bound m
- minimum cost function:

$$\sum_{i}^{n} f_m(a) - i \tag{3}$$

  where a is the array of cow positions, f_m is the order of wormhole we use (width less than w)
- A cow just needs to be able to swap to the corresponding position.
- proof.