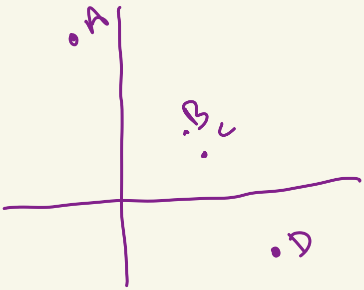
Exercise 1.1.1



All in one line

optimal → AB CD

Greedy → BC, AD

1.1.2

In the first step there are $\binom{n}{2}$ then $\binom{n-2}{2}$ and so on..

$$\frac{n!}{2!(n-2)!} \times \frac{(n-2)!}{(n-4)!2!} \cdots$$

$$\frac{n!}{2^{(n/2)}} \rightarrow \text{time complexity} = O(n^n)$$

1.1.3

Two-captcha

VPN

1.2.2

1) (b)

2) sliding window

3) (b)

4) (a)

5) (c)(d)

6) (a)

7) Number theory

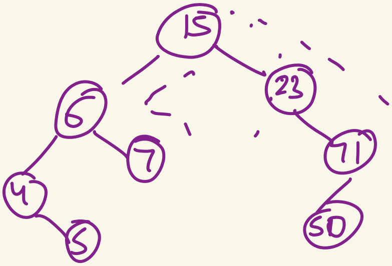
8)

9) Suffix Tree

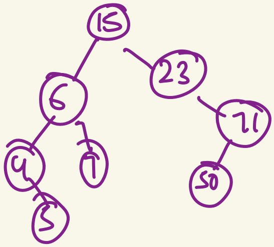
2.3.1) insertion & deletion = O(1)

2.3.2)

(1)



(2)



(3) findMin → left repeatedly

findMax → right repeatedly

(4) Yes

(5) right and then leftmost child

(6)

2.3.3) (1) Inorder Traversal

(2) O(n) approach

(3) O(n) approach

2.3.4) Yes, elements in sorted order

O(n) → inorder

2.3.5)

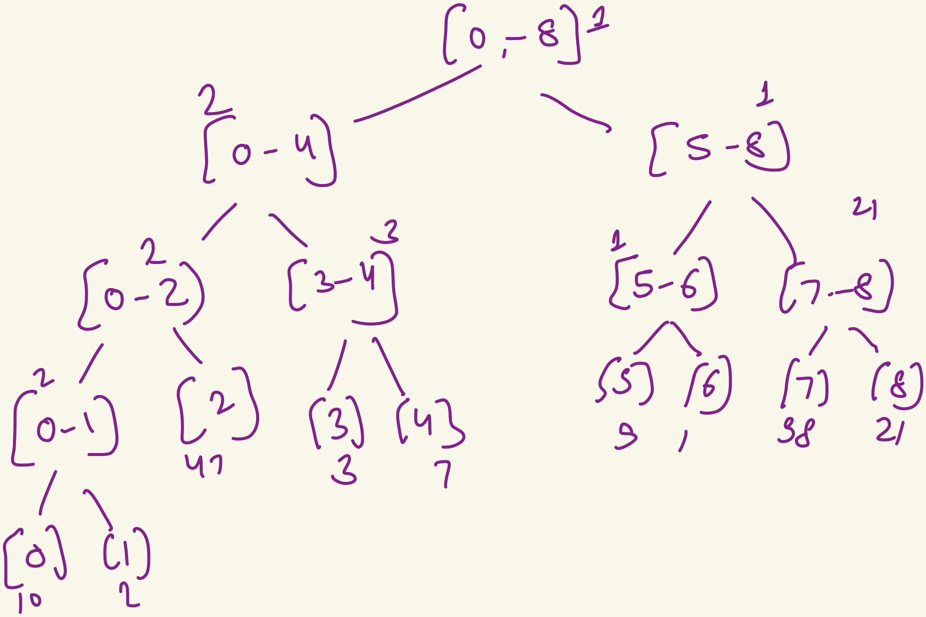
2.3.8) right child of root or child of left child.

2.3.9)

2.4.3-1

$A = \{10, 2, 47, 3, 7, 9, 1, 98, 21\}$

$\quad\;\; 0 \quad 1 \quad 2 \quad 3 \quad 4 \;\; 5 \;\; 6 \;\; 7 \quad 8$



2.4.3-2) store the sum instead of minimum

2.4.3-3) No, use DP

2.4.3-4) leaf node to root → update

2.4.3-5) Recreate segment tree

2.4.3-6) Yes, similar to searching for minimum of a range

2-Sat Problem

Each variable has two vertices in the implication graph, the variable itself and the negation/inverse of that variable. An edge connects one vertex to another if the corresponding variables are related by an implication in the corresponding 2-CNF formula.

Now, a 2-CNF is satisfiable if and only if there is no variable that belongs to the same $\underbrace{\text{SCC}}$ as its negation.

Strongly connected component.

Bitonic TSP

List of x-coordinates sorted by x-coordinates.

Find a tour that starts from the leftmost vertex, then goes strictly from left to right, and then upon reaching the rightmost vertex, the tour goes strictly from right to left. back to the starting vertex.

This tour behaviour is called bitonic

Solution: DP

For every vertex, whether it should be part of LR path or RL path.

Bracket Matching

Involves a question on whether a given set of braces is properly nested.

Solution → Use stack

Chinese Postman Problem

Also known as route inspection problem.

Handshaking Lemma → A non eulerian graph G must have an even number of vertices of odd degree.

subset of vertices of G that have odd degree = T

Create a complete graph K_n where n is the size of T.

An edge (i,j) in K_n has weight which is the shortest path weight of a path from i to j. form the vertices of K_n

Now, if we double the edges selected by the minimum weight perfect matching on this complete graph K_n, we will convert the non Eulerian graph G to another graph G' which is eulerian. Now find the Eulerian Tour.

Closest Pair Problems

Given a set of n points on a 2D plane, find two points with the closest Euclidean distance.

Naive Sol^n → $O(n^2)$

Divide and Conquer → $O(n \log n)$

Sort the points by x-coordinates and divide into two equal sets. (through a dividing line)

Single point in S → return ∞

Two points → return their euclidean distance.

d_1 = smallest distance in S_1

d_2 = " " in S_2

d_3 = min distance between S_1 and S_2 points

$d = \min(d_1, d_2, d_3)$

↳ Naive = $O(n^2)$

$d' = \min(d_1, d_2)$ A closer point in the right of the dividing line can only lie within a rectangle with width d' and height $2d'$.

Dinic's Algorithm

dist[v] = length of the shortest path from the source vertex s to v in the residual graph.

Edge (u,v) in the residual graph is included in the level graph L iff dist[v] = dist[u] + 1. A blocking flow is an s-t flow f such that after sending through flow f from s to t, the level graph L contains no s-t augmenting path anymore.

Formulas & Theorems

(1) Cayley's formula → there are n^{n-2} spanning trees of a complete graph with n labeled vertices

(2) Derangement → A permutation of the elements of a set such that none of the elements appear in their original position.

$D(n) = (D(n-1) + D(n-2)) \times (n-1)$ $\quad D(0)=1 \quad D(1)=0$

(3) Erdos Gallai's Thm: finite sequence to be the degree sequence of a simple graph:

$d_1 \geq d_2 \geq \dots \geq d_n$

iff $\sum_{i=1}^{n} d_i = \text{even}$

and $\sum_{i=1}^{k} d_i \leq k \times (k-1) + \sum_{i=k+1}^{n} \min(d_i, k)$

holds for $1 \leq k \leq n$

(4) Euler's formula for planar graph: $V - E + F = 2$

(5) Moser's circle → Number of pieces into which a circle is divided if n points on its circumference are joined by chords with no three internally concurrent. $g(n) = \binom{n}{4} + \binom{n}{2} + 1$

(6) Pick's theorem → i = number of integer points in the polygon
A = area of polygon $\quad$ b = integer points on boundary

$A = i + \frac{b}{2} - 1$

(7) number of spanning trees of a complete bipartite graph ($K_{m,n}$) = $m^{n-1} \times n^{m-1}$

Graph Matching

Select a subset of edges M of a graph G(V, E) so that no two edges share the same vertex.

Maximum cardinality matching → maximum number of matched edges

Perfect matching → + no unmatched vertex

Unweighted MCBM:

Max flow, Augmenting path, Hopcroft Karp's algorithm

Weighted MCBM:

reduce to mincost-max flow algorithm

Unweighted MCM:

Edmonds Algo or use DP with bitmask

Weighted MCM:

DP with bitmask.

Independent and Edge Disjoint Paths

→ max flow
no constraints on vertices

→ vertex disjoint paths from s to t

can be solved using max flow with both $v \in V$ and $e \in E$ having a capacity of 1

Inversion Index

Counting the minimum number of bubble sort swaps.

$O(n^2)$ → naive $O(n\log n)$ → Divide and conquer (merge sort kind of)

Josephus Problem

n people in a circle

Every k^{th} person is executed.

One person left.

$k = 2$: $\quad n = 1\, b_1 \ldots b_k$

person left $= b_1 b_2 \ldots b_k 1$

k: $\quad f(n,k) = \left(f(n-1, k) + k \right) \% \, n$

Kosaraji's Algorithm (for SCC)

Do DFS on the original graph and record finish time for each node.

Do DFS on the transposed graph (all edges reversed), considering nodes in decreasing order of their finish times

Each DFS traversal will give a SCC.

Lowest Common Ancestor

Naive sol^n → $O(n)$ go to root and then to second vertex

Reduce to RMQ and then use Sparse Table Data structure