Data Structures and Algorithms

Lecture 21

Aniket Basu Roy

2026-03-18 Wed

Contents

1 Agenda

1.1 Graphs

- Graph representations

- Special graph families: proximity graphs, intersection graphs
- k-Clique problem

2 Graph Representations

2.1 Adjacency List

- Array Adj of $|V|$ lists
- Adj$[u]$ contains all vertices v such that $(u, v) \in E$
- Space: $\Theta(V + E)$
- Time to check if $(u, v) \in E$: $O(\deg(u))$
- **Preferred for sparse graphs** ($|E| \ll |V|^2$)

2.2 Adjacency Matrix

- $|V| \times |V|$ matrix A where $A[u][v] = 1$ if $(u, v) \in E$, else 0
- Space: $\Theta(V^2)$
- Time to check if $(u, v) \in E$: $\Theta(1)$
- **Preferred for dense graphs** or when fast edge lookup is needed
- Undirected graphs: matrix is symmetric

2.3 Incidence Matrix

- $|V| \times |E|$ matrix B
- For undirected graphs: $B[v][e] = 1$ if vertex v is an endpoint of edge e, else 0
- For directed graphs: $B[v][e] = -1$ if e leaves v, $+1$ if e enters v, 0 otherwise

2.4 Summary

Representation	Space	Edge query	List neighbors
Adjacency list	$\Theta(V + E)$	$O(\deg(u))$	$\Theta(\deg(u))$
Adjacency matrix	$\Theta(V^2)$	$\Theta(1)$	$\Theta(V)$
Incidence matrix	$\Theta(VE)$	$O(E)$	$O(E)$

3 Proximity Graphs

Graphs defined on a **point set** $P \subset \mathbb{R}^2$ (or $\mathbb{R}^d$), where edges are determined by distances.

3.1 Euclidean Graph

- $V = P$, edge $\{p, q\}$ has weight $= \|p - q\|_2$
- Basis for geometric algorithms

3.2 Unit Disk Graph

- Connect p and q iff $\|p - q\| \leq r$ (for some fixed radius r)
- Models wireless communication networks (range $= r$)

3.3 k-Nearest Neighbor Graph

- Connect each point to its k closest points
- Directed: add edge from p to each of its k nearest neighbors
- Used in machine learning (kNN classifier), clustering

4 Intersection Graphs

Graphs where **vertices** = **objects** and **edges** = **pairs of objects that intersect** (or overlap).

4.1 Interval Graphs

- Each vertex = interval $[s_i, f_i]$ on the real line
- Edge $\{i, j\}$ iff intervals i and j overlap: $[s_i, f_i] \cap [s_j, f_j] \neq \emptyset$
- **Job scheduling connection**: intervals = jobs with start s_i and finish f_i
  - Two jobs conflict iff their intervals overlap
  - Maximum independent set in interval graph = maximum set of non-conflicting jobs
  - Solvable in polynomial time (greedy: sort by finish time)

4.2 Disk / Rectangle Intersection Graphs

- Vertices = disks (or rectangles) in the plane
- Edge iff two disks intersect
- Models frequency assignment, map labelling

5 k-Clique Problem

- Input: A simple undirected graph $G = (V, E)$
- Question: Does there exist a clique of size k in G?

A graph represented by it adjacency matrix will take $\Omega(n^k)$ time? Can we lower the time complexity if the graph is an unit disk graph?

6 Questions

1. For a graph with $|V| = n$ and $|E| = m$, what is the space complexity of an adjacency list representation? When is it better to use an adjacency matrix instead?
2. How many edges can a complete bipartite graph $K_{m,n}$ have?
3. A tree on n vertices has exactly $n - 1$ edges. Prove this.