Q9 — Community Detection (12 points)

Part 1 (6 points) — PPR Algorithm: Complexity & Why Sort?

Time Complexity

The PPR community detection algorithm has 4 sequential steps:

Step	Operation	Complexity
1	Pick seed set s	O(s) — constant
2	Run PPR with teleport = s	O(V) — also accept O(E) or O(V+E)
3	Sort nodes by PPR score	O(V) — given: linear-time sorting
4	Sweep to find min-conductance cut	O(V) — incremental conductance via memoization

All steps are sequential, so total worst-case complexity:

$$\boxed{O(V)}$$

(Saying "linear time" receives full credit. Express in terms of V, E, or s as appropriate.)

Why Sort Before Sweeping?

The sweep identifies the best cluster by finding the **prefix** of the node ordering with minimum conductance. For this to produce meaningful clusters, nodes with similar PPR scores (i.e., similar relevance to the seed) must appear **consecutively**.

Sorting by decreasing PPR score ensures that each prefix $\{v_1, v_2, \ldots, v_k\}$ contains the k nodes most closely associated with the seed set. Without sorting, the sweep would examine prefixes of arbitrarily ordered nodes, which have no structural relationship to community boundaries — the conductance values would be meaningless.

Sorting is necessary, not just helpful: the correctness of the sweep depends on processing nodes in order of their PPR scores. It is not merely an efficiency optimization.

Part 2 (3 points) — Louvain: Convergence & Optimality

Guaranteed to converge? Yes.

Each iteration moves a node to a different community only if modularity **strictly increases** (

$\Delta Q > 0$). Since:

1. Modularity is **bounded** ($Q \in [-1, 1]$)
2. Each move **strictly increases** Q
3. There are **finitely many** possible community assignments

...the algorithm cannot cycle and must terminate in finite steps.

Guaranteed to reach maximum modularity? No.

Louvain is a **greedy** algorithm — it only considers local, single-node moves. It converges to a **local maximum** of modularity, not necessarily the global one. Reaching the global optimum might require simultaneously reassigning multiple nodes (temporarily decreasing Q), which the greedy strategy never allows.

Part 3 (3 points) — Can NN-Descent Replace PPR/Louvain?

Answer: No.

NN-Descent and community detection algorithms solve **fundamentally different problems**:

Algorithm	Role	Input	Output
NN-Descent	Graph construction	Data points + similarity	K-NN graph
PPR / Louvain	Community detection	Graph	Partition into communities

NN-Descent efficiently builds an approximate K-nearest-neighbor graph from raw data. It has **no mechanism** for defining or optimizing community quality — no conductance computation, no modularity optimization, no partition output.

These are **complementary pipeline stages**, not interchangeable alternatives:

$$\text{Raw data} \xrightarrow{\text{NN-Descent}} \text{K-NN graph} \xrightarrow{\text{PPR / Louvain}} \text{Communities}$$

After building the K-NN graph with NN-Descent, one still needs PPR or Louvain to identify the actual communities within that graph.