MCP SERVER | NO CODE | CLOUD HOSTED

Minimum Spanning Tree Calculator MCP for AI Agents

Calculating Optimal Network Connectivity and Graph Structures

The Minimum Spanning Tree Calculator figures out the most efficient network connections in a graph. It uses established algorithms like Kruskal's and Prim's to find both minimum-weight (cheapest) and maximum-weight paths connecting all nodes without forming any loops. This MCP helps engineers model complex networks, from electrical grids to data flows, by identifying optimal edge sets.

A+ Quality Score 100/100

mst | kruskal | prim | graph | algorithm

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JULY 2026
vinkius.com/mcp/minimum-spanning-tree-calculator

SCAN TO VIEW

The connectivity layer between AI and the world's software.

Vinkius sits between AI and every application. All communication passes through Vinkius Cloud via the Model Context Protocol (MCP) — with governance, observability, and security at every layer.

Your AI Connections Run Through Vinkius Cloud

The world's largest managed MCP catalog

Vinkius is the connectivity layer where AI connects to the software your business already runs. We handle the hosting, the security, the credentials, the uptime — you get agents that actually do things.

We operate the world's largest managed MCP catalog. Major SaaS platforms, CRMs, databases, and cloud providers — running, monitored, production-ready. This MCP server is hosted and maintained by the Vinkius Cloud for AI Agents.

> *The agent doesn't manage credentials, doesn't manage uptime, doesn't manage security. Vinkius does.*
>
> — Architecture principle

Four Pillars of the Vinkius Runtime

01 — Security by design

Credentials stay encrypted at rest via AES-256. The AI agent never touches raw keys — they're injected into a sandboxed V8 isolate at runtime. Actions are logged, and connections have an emergency kill switch.

02 — Built on MCP Fusion

This MCP server was built with **MCP Fusion**, the open-source framework (Apache 2.0) that powers the entire Vinkius catalog. Schema-as-firewall strips undeclared fields, compiled PII redaction runs at zero overhead, and cryptographic lockfiles produce git-diffable audit trails.

03 — Deterministic observability

Eight immutable metrics per endpoint: request volume, p95 latency, error rate, active connections, cost attribution. A live payload feed logs every tool call with mutation detection.

04 — Autonomous operations

Servers are deployed, monitored, and patched autonomously. New capabilities and security patches ship weekly. Zero-downtime deployments ensure continuous availability across all managed MCP servers.

AES-256	Ed25519	24h TTL	V8 Isolate
Encryption at rest	PKI vault signatures	Ephemeral session keys	Sandboxed execution

One Token. Instant Access.

Every MCP server on Vinkius is accessed through a **Connection Token**. Tokens are generated in the cloud dashboard and produce a unique MCP endpoint URL. Paste this URL into any MCP-compatible client — no SDK required.

A single token can serve **multiple AI clients simultaneously**, or you can issue separate tokens per client for granular access control. Each token tracks its own request count, last activity timestamp, and can be individually enabled or revoked.

MCP ENDPOINT

```
https://edge.vinkius.com/{token}/mcp
```

Claude · Cursor · VS Code · Windsurf · Grok · Gemini

Security Is the Architecture

Security in Vinkius is not a feature — it's the foundation of the runtime. The gateway enforces multiple independent protection layers between AI agents and third-party APIs.

01 — Ed25519 PKI Vault

Every workspace has an Ed25519 Master Key. Session keys are generated ephemerally (24h TTL) and signed by the Master Key. Credentials never leave the vault boundary.

02 — V8 Isolate Sandboxing

Tool code runs inside isolated-vm V8 isolates with 64 MB memory caps and per-request timeouts. No filesystem access, no network access except through the SSRF-guarded fetch bridge.

03 — SSRF Guard

All outbound HTTP requests are DNS-resolved and validated before execution. Private IP ranges (10.x, 172.16-31.x, 192.168.x, AWS metadata 169.254.x) are blocked at the network layer.

04 — DLP & PII Redaction

A ResponseGuard pipeline intercepts every tool response. Configurable redaction patterns strip sensitive fields (emails, SSNs, card numbers) before data reaches the AI agent.

05 — Cryptographic Audit Trail

Every request is signed into a SHA-256 hash chain with Ed25519 signatures. Events form a tamper-proof, SIEM-exportable forensic record.

06 — Honeytoken Trap System

Phantom credentials are injected into isolated environments. If a honeytoken is used outside Vinkius infrastructure, the server is quarantined instantly.

Emergency Kill Switch

EU AI Act Art. 14(1) Compliant

The kill switch is an **emergency halt** mechanism — not a simple toggle. When triggered, it executes three actions atomically:

01 — Server deactivated

The MCP server is immediately taken offline across the entire cluster.

02 — All tokens revoked

Every connection token is invalidated. Total lockout — reconnection blocked until new tokens are issued.

03 — WebSocket connections killed

Active connections terminated via Redis pubsub broadcast. Propagates to every runtime node in the cluster.

Full Visibility. Zero Guesswork.

The Vinkius cloud dashboard includes a full MCP Governance suite — real-time analytics and security controls for production AI operations.

Control Plane

KPI dashboard with request volume, latency, success rate, token consumption, and AI-generated operational briefings.

Agent Activity

Which AI clients are connecting, how often, and what they're doing — real-time session tracking.

FinOps

Cost tracking per tool, payload compression savings, budget optimization signals, and consumption trends.

Tool Health

Slowest and most error-prone tools, with actionable root-cause insights and performance baselines.

Firewall & DLP

PII redaction activity, sensitive data protection counters, and security event timeline.

Incident Log

Error trends, failure rates, status-code breakdowns, and forensic audit trail access.

> Get started at **cloud.vinkius.com** — connect your AI agent in under 60 seconds.

Minimum Spanning Tree Calculator MCP

3 tools available
Cloud-hosted on Vinkius

When you're designing a network or analyzing connectivity, figuring out the most efficient path isn't always obvious. You need to connect every point while keeping costs low, or sometimes, maximizing capacity. This MCP gives your AI agent the tools to handle that math. Instead of manually testing thousands of edge combinations, your agent runs the calculations instantly. For instance, you can run `compute_kruskal_mst` to find the minimum cost required to connect all nodes in a given graph. If you need to start building from one specific point, `compute_prim_mst` grows the network outward using only the best available connections. It also handles the opposite case with `compute_max_spanning_tree`, maximizing connection weight when needed. Because this MCP is part of Vinkius's catalog, your agent connects once to access these critical graph algorithms and more.

Core Capabilities

01 — Calculate Minimum Spanning Trees (MST)

Finds the set of lowest-weight edges that connect all points in a network without creating any cycles.

02 — Identify Maximum Spanning Trees

Determines the highest-weight connections possible across a graph, ignoring cycle constraints.

03 — Grow Networks from a Specific Node

Uses Prim's algorithm to build an MST starting at a designated node by always choosing the cheapest next connection.

One Click on Vinkius — From Prompt to Execution

Available at **vinkius.com/mcp/minimum-spanning-tree-calculator** — connect your AI agent in three steps.

01 You provide your agent with an edge list, detailing all nodes and the connecting weights (costs) between them.

02 Your agent selects which algorithm you need—Kruskal's for overall minimum cost, Prim's starting from a point, or maximum weight calculation.

03 The MCP processes the graph data and returns the specific set of edges that form the optimal tree structure, along with the total calculated weight.

The bottom line is that you input your network map, and the MCP outputs the mathematically proven most efficient set of connections for whatever goal you define—minimum cost or maximum capacity.

Built For

This tool targets data scientists, civil engineers, and optimization analysts. If your job requires modeling infrastructure or resource allocation based on connectivity costs, this MCP is essential. It handles the core math so you can focus on solving real-world physical problems.

Civil Engineer

Modeling new utility lines (water pipes, fiber optic cables) to ensure every point gets connected using the minimum amount of piping material.

Data Scientist

Analyzing network traffic data or social graphs to find critical paths and dependencies within a dataset.

Operations Analyst

Optimizing logistics routes or electrical grid designs to ensure full coverage with the lowest possible material cost.

What Changes When You Connect

01 Pinpoint minimum infrastructure costs. Instead of guessing, use `compute_kruskal_mst` to find the absolute lowest-cost set of connections needed for your network.

02 Optimize from a single source. If you know where construction starts, let `compute_prim_mst` guide you by growing the tree outward using only the best available connections.

03 Model maximum capacity. When maximizing data throughput or power lines is key, `compute_max_spanning_tree` finds the most robust connection set.

04 Eliminate redundant testing. You stop wasting time manually checking combinations; your agent handles the complex graph theory math instantly.

05 Handle varying topologies. This MCP knows if your result is a single connected tree or a collection of separate, unconnected components (a spanning forest).

Real-World Applications

Designing a Low-Cost Electrical Grid

A city planner needs to run power lines across a new development site. Asking their agent to use the Minimum Spanning Tree Calculator finds the cheapest set of routes that still guarantees every home gets electricity, preventing costly overspending on unused connections.

Optimizing Fiber Optic Laying

A telecom company must lay new fiber optic cables connecting dozens of buildings. Using Prim's algorithm helps them plan the initial segment from their central hub, ensuring they build out efficiently without crossing redundant routes.

Analyzing Data Dependencies in a System

A data architect needs to know the minimum set of connections required between microservices. The agent runs Kruskal's method on the service dependency graph, identifying the bare minimum links needed for system functionality.

Maximizing Warehouse Connectivity

The facility manager wants to link several storage zones with new high-capacity conveyor belts. Using the Maximum Spanning Tree finds the configuration that maximizes total throughput capacity across all connected points.

Patterns to Avoid

Manually checking every edge combination

✗ AVOID

Trying to find the cheapest path by drawing lines on a map and manually comparing costs, which is impossible for graphs with more than ten nodes.

✓ INSTEAD

Feed all your node data and weights into this MCP. Use `compute_kruskal_mst` to instantly verify the absolute minimum cost set of edges that connect everything.

Confusing MST with shortest path

✗ AVOID

Assuming the cheapest connection between two adjacent nodes is enough; forgetting that you need a continuous path connecting ALL nodes.

✓ INSTEAD

Remember, the goal isn't just A to B. Use this MCP to find the full spanning structure. Algorithms like Prim's ensure connectivity across the entire set of points.

Only checking for minimum weight

✗ AVOID

Stopping at the cheapest connection without considering that a high-capacity link might be required elsewhere in the network.

✓ INSTEAD

If maximizing strength or bandwidth is necessary, use `compute_max_spanning_tree` to find the optimal configuration based on weight maximization.

The Right Fit

Use this MCP when your primary goal involves connectivity optimization across a defined set of nodes. If you need to know the absolute lowest cost way to connect everything, use `compute_kruskal_mst`. If the connection must originate from one specific point (like a central power station), then Prim's algorithm via `compute_prim_mst` is better. Only use this tool if your problem can be modeled as finding an optimal subgraph that connects all vertices without cycles. Don't use it if you are simply looking for the shortest path between two points, because that requires different tools entirely. For pure weight maximization, rely on `compute_max_spanning_tree`.

Minimum Spanning Tree Calculator: Optimizing Infrastructure Costs with Graph Theory

Today, infrastructure planning involves tedious manual work. Engineers map out potential routes for utilities or data cables and then spend hours comparing costs across different segments. They might copy weights into spreadsheets, running basic calculations that only confirm local connections but fail to guarantee minimum cost coverage across the entire network.

With this MCP, you simply provide your agent with the graph's edge list. It runs `compute_kruskal_mst` and immediately identifies the absolute lowest-weight set of edges required. You get the definitive map that guarantees full connectivity at the lowest possible expenditure.

Minimum Spanning Tree Calculator: Maximizing Connectivity Weight in Network Design

Sometimes, minimizing cost isn't the goal; maximizing capacity is. Manual analysis often overlooks high-weight connections that could be critical for data flow or power redundancy, settling instead for simpler, cheaper paths.

By using `compute_max_spanning_tree`, your agent models the system to ensure maximum possible connectivity weight. It provides a robust network design that anticipates peak loads and maximizes resource utilization.

Minimum Spanning Tree Calculator: 3 Graph Algorithms for Optimization

Use these tools to calculate minimum, maximum, or point-sourced spanning trees from complex edge lists.

#	TOOL	DESCRIPTION
01	`compute_kruskal_mst`	Calculates the Minimum Spanning Tree using Kruskal's algorithm on a provided edge list.
02	`compute_max_spanning_tree`	Determines the Maximum Spanning Tree weight and edges from an input graph.
03	`compute_prim_mst`	Calculates the Minimum Spanning Tree using Prim's algorithm, starting from a specified node.

See It in Action

Real prompts you can use once this MCP is connected to your AI agent through Vinkius Cloud.

U We need to connect five remote sensor nodes using new wireless relays. The cost data is attached. What's the minimum relay setup?

Minimum Spanning Tree Results

- **Total Minimum Weight:** 45 units
- **Optimal Edges Found:**

* (Node A, Node B) - Cost: 12
* (Node C, Node D) - Cost: 8
* (Node A, Node C) - Cost: 10
* (Node E, Node B) - Cost: 15

This set of relays guarantees all nodes are connected using the least amount of material.

U Starting from the main control hub (Hub Alpha), what's the most efficient way to connect the three adjacent sub-stations? Use Prim's method.

Minimum Spanning Tree (Prim's) Starting at Hub Alpha

- **Connection Sequence:**

1. (Hub Alpha, Sub-Station Beta) - Cost: 5
2. (Sub-Station Gamma, Sub-Station Beta) - Cost: 3
3. (Sub-Station Delta, Sub-Station Gamma) - Cost: 7

- **Total Path Weight:** 15 units

The agent confirms that this sequence provides the lowest total cost while guaranteeing a continuous path from your specified starting point.

U We have three major data links with weights of 20, 30, and 10. What is the maximum capacity connection set we can build?

Maximum Spanning Tree Calculation

• The highest connectivity set uses edges:
1. Link A-B (Weight: 30)
2. Link B-C (Weight: 20)

• **Total Maximum Weight:** 50 units

The agent confirms that by selecting these two links, you achieve the highest possible total capacity without creating a redundant loop.

Frequently Asked Questions

01 **How does the Minimum Spanning Tree Calculator help with network design?**

It solves for the minimum cost way to connect every single point in your network. Instead of guessing, it uses proven algorithms to find the absolute cheapest set of required connections, saving you time and money on materials.

02 **Do I use this MCP if my goal is maximizing capacity instead of minimizing cost?**

Yes, but you need a specific tool. If your priority is maximum throughput or strength (like power lines), use the Maximum Spanning Tree feature to find the highest-weight connection set.

03 **What's the difference between Kruskal's and Prim's algorithms in this MCP?**

Kruskal's looks at all edges globally to build the cheapest network possible. Prim's is better if you have a specific starting point, as it builds the optimal network outward from that single node.

04 **Can I use this MCP for anything other than cables and power lines?**

Absolutely. This tool handles any graph structure with weighted edges. You can apply it to data dependencies, logistics routes, or sensor networks—anything where connectivity matters.

05 **Does the Minimum Spanning Tree Calculator handle multiple separate groups of nodes?**

Yes. If your network is naturally divided into unconnected sections, the MCP will identify that the result is a spanning forest, showing you all necessary connections for every isolated group.

Go Live in 60 Seconds

Get your connection token from **cloud.vinkius.com**, then paste the endpoint URL into any MCP-compatible client.

YOUR MCP ENDPOINT

```
https://edge.vinkius.com/[TOKEN]/mcp
```

CLIENT	WHERE TO CONFIGURE
Claude AI	Profile → Customize → Connectors → "+" → Add custom connector → Paste endpoint
Cursor	Settings → Features → MCP Servers → "+ Add New MCP Server" → Type: SSE → Paste endpoint
VS Code	Ctrl/Cmd+Shift+P → "MCP: Add Server" → add `"minimum-spanning-tree-calculator": { "url": "…" }`
Windsurf	MCP Settings → `mcp_settings.json` → Add endpoint URL
ChatGPT	Settings → Tools & plugins → Add MCP server → Paste endpoint
Gemini	Extensions → Add MCP Server → Paste endpoint URL

ASK AN AI ABOUT THIS

Let your preferred AI explain this MCP server

- Ask ChatGPT
- Ask Claude
- Ask Perplexity
- Ask Gemini
- Ask Grok

READY TO CONNECT

Minimum Spanning Tree Calculator is live on Vinkius Cloud.

Get your connection token, paste it into your AI agent, and start building. No SDK. No deployment. Just results.

Start at cloud.vinkius.com →

vinkius.com · support@vinkius.com

INDEPENDENT PLATFORM DISCLAIMER

Vinkius is an independent platform and is not affiliated with, endorsed by, sponsored by, verified by, or otherwise authorized by Minimum Spanning Tree Calculator. All third-party trademarks, logos, and brand names are the property of their respective owners. Their use in this document is strictly for informational purposes to identify service compatibility and interoperability.

DOCUMENT INFORMATION

Generated	July 2026
MCP Server	Minimum Spanning Tree Calculator MCP
Server ID	`019f2d2d-5f48-71dc-98d3-f4c914bca8e5`
Platform	Vinkius Cloud for AI Agents
Endpoint	`https://edge.vinkius.com/{token}/mcp`

LICENSE & USAGE

This document is generated automatically by the Vinkius PDF Engine. Content reflects the MCP server configuration at the time of generation and may change as updates are deployed. For the most current information, visit vinkius.com/mcp/minimum-spanning-tree-calculator.