

รายงาน

เรื่อง: หลักการทำงานของ Dijkstra's Algorithm พร้อมตัวอย่างการประยุกต์ใช้งานในระบบเครือข่าย

บทนำ

ในระบบเครือข่ายคอมพิวเตอร์ การค้นหาเส้นทางที่สั้นที่สุดระหว่างต้นทางและปลายทางเป็นสิ่งสำคัญ เพื่อให้การรับส่งข้อมูลมีประสิทธิภาพสูงสุด Dijkstra's Algorithm เป็นหนึ่งในอัลกอริธึมที่นิยมใช้กันอย่างแพร่หลาย โดยเฉพาะในโปรโตคอลที่ใช้หลักการ Link-State Routing เช่น OSPF (Open Shortest Path First) ซึ่งสามารถคำนวณเส้นทางที่ดีที่สุดจากเราเตอร์หนึ่งไปยังทุกเราเตอร์ในเครือข่าย

หลักการทำงานของ Dijkstra's Algorithm

Dijkstra's Algorithm เป็นอัลกอริธึมที่ใช้ในการหาเส้นทางที่สั้นที่สุดในกราฟที่มีน้ำหนักบวกทั้งหมด (non-negative weights) โดยมีหลักการทำงานดังนี้:

ขั้นตอนการทำงาน

เริ่มต้น: กำหนดให้ระยะทางจากต้นทางไปยังตัวเองเป็น 0 และระยะทางไปยังโหนดอื่น ๆ เป็น ∞ (Infinity)

สร้างชุดของโหนดที่ยังไม่ได้เยี่ยมชม (unvisited)

เลือกโหนดในชุด unvisited ที่มีระยะทางน้อยที่สุด

สำหรับโหนดนั้น:

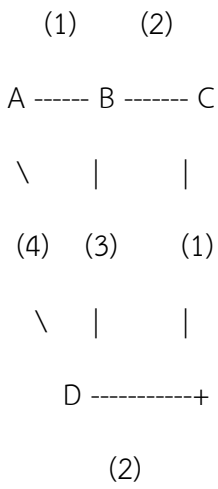
ตรวจสอบเพื่อนบ้านที่เชื่อมต่ออยู่

หากเส้นทางผ่านโหนดปัจจุบันไปยังเพื่อนบ้านมีระยะทางน้อยกว่าเส้นทางเดิม ให้ทำการอัปเดต

ทำซ้ำจนกว่าโหนดทั้งหมดจะถูกเยี่ยมชมครบ

ตัวอย่างประกอบ

สมมติว่ามีกราฟดังนี้:



จุดเริ่มต้น: A

ผลลัพธ์ของ Dijkstra's Algorithm จาก A ไปยังโหนดอื่นๆ:

ปลายทาง ระยะทางรวม เส้นทางผ่าน

A	0	A
B	1	A → B
C	3	A → B → C
D	4	A → D
E	5	A → B → E

ตัวอย่างการใช้งานในระบบเครือข่าย

ใช้ในโปรโตคอล OSPF (Open Shortest Path First)

OSPF เป็นโปรโตคอลประเภท Link-State Routing

เราเตอร์แต่ละตัวจะสร้างแผนที่ของเครือข่ายทั้งหมด (Topology Map)

จากนั้นใช้ Dijkstra's Algorithm คำนวณเส้นทางที่สั้นที่สุดจากตนเองไปยังเราเตอร์ทุกตัว

เส้นทางที่ได้จะถูกใช้ใน Routing Table เพื่อกำหนดวิธีการส่งข้อมูลในระบบเครือข่าย

การใช้งานใน Software Defined Networking (SDN)

ในระบบ SDN ที่ใช้ Centralized Controller

ตัว Controller ใช้ Dijkstra's Algorithm ในการเลือกเส้นทางการส่งข้อมูลจากต้นทางไปยังปลายทาง

ช่วยบริหาร traffic และลดความแออัด

การใช้งานในศูนย์ข้อมูล (Data Center)

ใช้สำหรับคำนวณเส้นทางที่เร็วที่สุดระหว่าง Server, VM หรืออุปกรณ์ต่าง ๆ ภายในศูนย์ข้อมูล

ข้อดีและข้อจำกัด

ข้อดีของ Dijkstra

คำนวณเส้นทางที่สั้นที่สุดได้แม่นยำ

เหมาะกับกราฟขนาดใหญ่ที่ไม่มีน้ำหนักติดลบ

ทำงานเร็ว โดยเฉพาะเมื่อใช้ร่วมกับโครงสร้างข้อมูลแบบ Priority Queue

ข้อจำกัดของ Dijkstra

ไม่สามารถใช้กับกราฟที่มีน้ำหนักติดลบ

ต้องรู้ Topology ทั้งหมดของกราฟก่อนล่วงหน้า

ต้องใช้หน่วยความจำมากเมื่อกราฟมีขนาดใหญ่

สรุป

Dijkstra's Algorithm เป็นอัลกอริธึมที่มีประสิทธิภาพและแม่นยำในการคำนวณเส้นทางที่สั้นที่สุดในกราฟที่มีน้ำหนักบวก โดยเฉพาะในระบบเครือข่าย เช่น OSPF หรือ SDN ที่ต้องใช้การคำนวณเส้นทางผ่านโหนดจำนวนมาก อัลกอริธึมนี้ยังสามารถนำไปประยุกต์ใช้ในงานอื่น ๆ เช่น ระบบนำทาง GPS การวางแผนเส้นทางของหุ่นยนต์ หรือเกมที่ต้องมีการคำนวณ Pathfinding