Overzicht algoritmes: grafen en kortste paden

Compact schema van de algoritmes die tot nu toe aan bod kwamen. Notatie: |V| = aantal knopen, |E| = aantal bogen/edges.

1. Basis traversals

Algoritme	Waarvoor?	Soort grafen	Hoe werkt het kort?	Complexiteit
DFT / DFS	Graaf doorlopen; basis voor componenten, cycli, topologische sortering en lowlink.	Gericht en ongericht.	Ga zo diep mogelijk. Bij terugkeren/sluiten kan je extra info verzamelen, zoals sluitingsvolgorde of lowlink.	O(|V| + |E|)
BFT / BFS	Graaf doorlopen; kortste paden in ongewogen grafen.	Gericht en ongericht.	Werk laag per laag met een queue. In een ongewogen graaf is laagnummer = afstand vanaf de bron.	O(|V| + |E|)

2. Connectiviteit

Algoritme	Waarvoor?	Soort grafen	Hoe werkt het kort?	Complexiteit
Connected components	Componenten vinden.	Ongerichte grafen.	Start DFS/BFS vanaf een nog niet bezochte knoop. Alles wat je bereikt zit in dezelfde component. Herhaal tot alles bezocht is.	O(|V| + |E|)
Union-Find	Dynamisch bijhouden welke knopen in dezelfde component zitten.	Meestal ongericht.	Elke component is een set. Union voegt sets samen; Find zegt in welke set een knoop zit. Handig bij Kruskal.	Bijna O(1) per operatie; formeel O(alpha(n))
Bruggen	Bogen vinden waarvan verwijderen de graaf splitst.	Ongerichte grafen.	DFS met pre[v] en low[v]. Voor tree-edge v-w is de edge een brug als low[w] > pre[v].	O(|V| + |E|)
Articulatiepunten	Knopen vinden waarvan verwijderen de graaf splitst.	Ongerichte grafen.	Zelfde lowlink-idee als bruggen. Een knoop is kritiek als een kind-subboom niet terug kan naar een voorouder.	O(|V| + |E|)
Bipartiet testen	Nagaan of een graaf 2-kleurbaar is.	Ongerichte grafen.	BFS/DFS en buren altijd tegengestelde kleur geven. Conflict betekent: niet bipartiet.	O(|V| + |E|)

3. Minimale opspannende boom

Algoritme	Waarvoor?	Soort grafen	Hoe werkt het kort?	Complexiteit
Kruskal	Minimum spanning tree.	Gewogen ongerichte grafen.	Sorteer bogen op gewicht. Voeg een boog toe als ze geen cyclus maakt. Union-Find test snel of dat mag.	O(|E| log |V|)
Prim-Jarnik	Minimum spanning tree.	Gewogen ongerichte grafen.	Groei een MST vanuit een startknoop. Priority queue bewaart voor elke buitenknoop zijn goedkoopste kandidaatboog naar de MST.	O((|V| + |E|) log |V|)
Boruvka	Minimum spanning tree.	Gewogen ongerichte grafen.	Elke component kiest zijn goedkoopste uitgaande boog. Voeg die toe en merge componenten. Herhaal.	Meestal O(|E| log |V|)

Overzicht algoritmes: grafen en kortste paden

4. Gerichte grafen

Algoritme	Waarvoor?	Soort grafen	Hoe werkt het kort?	Complexiteit
Topologisch sorteren met DFS	Volgorde van afhankelijkheden vinden.	DAGs.	Doe DFS. Bij sluiten van een knoop zet je die vooraan in de lijst. Omgekeerde sluitingsvolgorde is topologisch.	O(|V| + |E|)
Kahn	Topologisch sorteren.	DAGs.	Bereken in-degrees. Start met knopen met in-degree 0. Verwijder ze en verlaag in-degrees van hun uitgaande buren.	O(|V| + |E|)
Kosaraju	Strongly connected components vinden.	Gerichte grafen.	DFS op originele graaf en sluitingslijst maken. Keer alle bogen om. DFS opnieuw in die lijstvolgorde; elke DFS-boom is een SCC.	O(|V| + |E|)
Tarjan SCC	Strongly connected components vinden.	Gerichte grafen.	Een DFS met stack, preorder en lowlink. Als lowlink[v] = pre[v], dan is v root van een SCC; pop stack tot v.	O(|V| + |E|)
Warshall	Transitieve sluiting: bestaat er een pad u -> v?	Gerichte grafen.	Matrix met bereikbaarheid. Voor elke tussenknoop k: zet M[i][j] true als i naar k kan en k naar j kan.	O(|V|^3)

5. Kortste paden en planning

Algoritme	Waarvoor?	Soort grafen	Hoe werkt het kort?	Complexiteit
BFS shortest path	Single-source shortest path.	Ongewogen grafen.	Elke boog telt als gewicht 1. BFS-lagen geven de kortste afstand vanaf de bron.	O(|V| + |E|)
Bellman-Ford	Single-source shortest path.	Gewogen grafen; negatieve gewichten mogen; geen negatieve cyclus bereikbaar vanaf start.	Init start = 0 en rest = oneindig. Herhaal |V|-1 keer: relaxeer alle bogen. Extra ronde detecteert negatieve cycli.	O(|V| * |E|)
Dijkstra	Single-source shortest path.	Gewogen grafen met gewichten ≥ 0.	Kies telkens de nog niet definitieve knoop met kleinste voorlopige afstand. Relaxeer zijn uitgaande bogen.	O((|V| + |E|) log |V|)
Lawler	Single-source shortest path in een DAG.	Gewogen DAGs; negatieve gewichten mogen.	Topologisch sorteren. Ga knopen in die volgorde af en relaxeer uitgaande bogen. Elke afstand wordt meteen definitief.	O(|V| + |E|)
Floyd-Warshall	All-pairs shortest path.	Gewogen gerichte grafen; negatieve gewichten mogen, maar geen negatieve cycli.	Afstandsmatrix D. Voor elke tussenknoop k: check of i -> k -> j korter is dan i -> j.	O(|V|^3)
PERT / kritisch pad	Vroegste eindtijd en kritische taken in projectplanning.	Gewogen DAGs.	Bogen zijn taken en gewichten zijn duurtijden. Bereken langste pad met max in plaats van min; een taak kan pas starten als alle voorgangers klaar zijn.	O(|V| + |E|)

Kernonderscheid: Warshall geeft alleen bereikbaarheid true/false; Floyd-Warshall geeft kortste afstanden. PERT gebruikt een langste pad in een DAG, omdat alle voorafgaande taken klaar moeten zijn.