

Datastructuren

November 2, 2009

1. What is the relationship between the sum of the degrees of all vertices and number of edges of graph $G = (V, E)$?
2. What is the complexity of the breadth-first search algorithm?
3. Show that a simple graph is connected if it has a spanning tree. (A graph $G = (V, E)$ is *simple* iff $G = (V, E)$ consists of a nonempty set V of *vertices* and possibly an empty set E of *edges*, each edge being a set of two vertices from V .)
4. Show that a tree with n vertices has $n - 1$ edges.
5. Consider the graph in Figure 1 Construct the bsf-tree starting with node 7. What is the bsf-forest for the whole graph.
6. Design an algorithm to determine the connected components of a graph.
7. Consider the graph in Figure 1 Determine whether this graph is bipartite. If so provide a coloring.
8. Consider the DAG in Figure 2 Determine how many topological orderings this DAG can have.
9. Geef van onderstaande grafen een *adjacency matrix* en een *adjacency list* representatie.

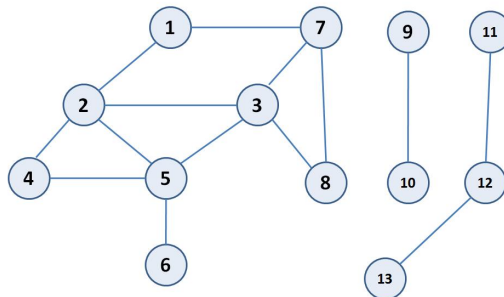
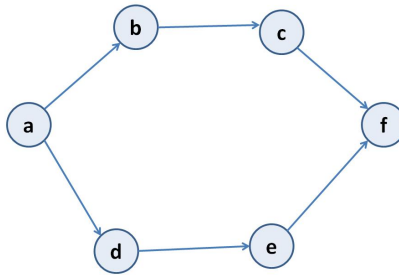
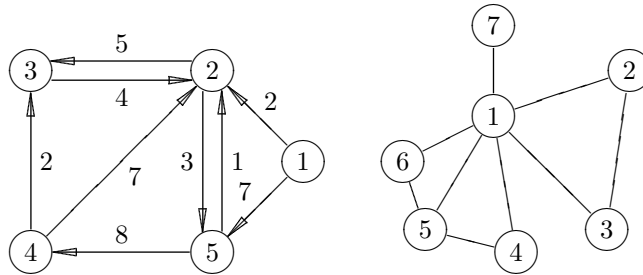


Figure 1: a graph

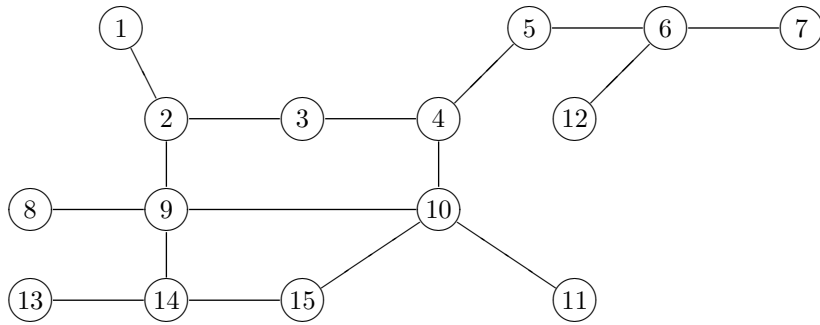


How many topological orderings does this DAG have?

Figure 2: a Directed Acyclic Graph



10. Bepaal van deze grafen tevens de opspannende bomen voor de *breadth-first* en *depth-first* wandelingen. Geef ook de volgorde aan waarin de knopen bezocht worden.
11. Beschouw de volledige graaf op n knopen. Hoe ziet de opspannende boom van deze graaf er uit voor elk van de beide graafwandelingen ?
12. Implementeer de *depth-first* graafwandeling met behulp van een stapel (dus zonder gebruik te maken van recursie).
13. **a.** Door bij *depth-first search* niet gewoon de nieuwe knopen op de stapel te zetten, maar ze steeds te voorzien van informatie die hun 'ouder' aangeeft (dwz. de knoop die bezocht werd toen de nieuwe knoop op de stapel kwam) kunnen we op eenvoudige wijze uitprinten hoe de *dfs*-opspannende boom van de graaf er uitziet. Leg dit uit.
b. Bekijk onderstaande ongerichte graaf. Geef de volgorde aan waarin de knopen bij een *depth-first* wandeling bezocht worden. Welke knopen staan er op de stapel wanneer we knoop 9 bezoeken? Welke knopen zijn de ouders van de knopen op de stapel?



c Hoe kan het *dfs*-algoritme aangepast worden om te testen of een ongerichte graaf een kring bevat (dwz. geen boomstructuur heeft) ?

Hoe kunnen we zo'n kring ook op eenvoudige wijze afdrucken?

14. Schrijf een algoritme dat, voor een gegeven knoop in een graaf, een knoop op maximale afstand in de graaf oplevert. Hierbij meten we de afstand tussen twee knopen als het minimale aantal takken van paden tussen beide knopen.
15. Geef een algoritme dat, voor een gegeven gerichte graaf, bepaalt of er een knoop is van waaruit alle andere knopen bereikbaar zijn, en — als dit het geval is — deze knoop uitvoert.