

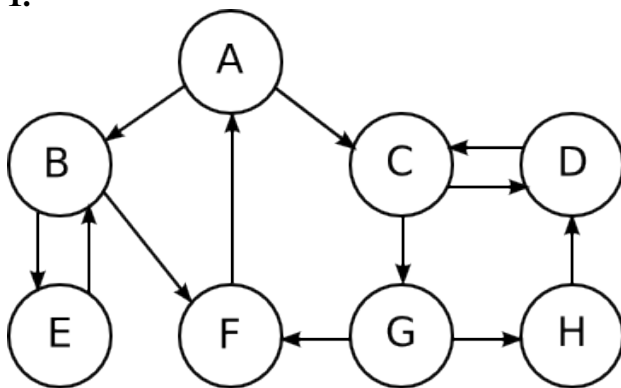
# Lösningförslag till SI-möte #13

## Programkonstruktion och datastrukturer

Elias Castegren & Kristiina Ausmees

elca7381@student.uu.se || krau6498@student.uu.se

1.



i) I vilken ordning besöks noderna i en Breadth-First-Search (BFS)?

*Om man till exempel börjar på nod A och väljer vägar motsols får man ordningen  $\langle A, B, C, E, F, G, D, H \rangle$*

*Se föreläsningsslides för metod*

ii) I vilken ordning besöks noderna i en Depth-First-Search (DFS)?

*Om man börjar på nod A och väljer vägar motsols besöker man noderna i ordningen  $\langle A, B, E, F, C, G, H, D \rangle$*

*(notera att "finishing order" är  $\langle E, F, B, D, H, G, C, A \rangle$ )*

*Se föreläsningsslides för metod*

iii) Är grafen ett träd?

*Nej, grafen innehåller cykler*

iv) Är grafen en komponent?

*Ja, man kan nå alla noder från vilken nod som helst*

v) Finns det någon väg som besöker alla noder exakt en gång?

*Ja, t.ex.  $\langle E, B, F, A, C, G, H, D \rangle$*

2. –

3.

	Best	Average	Worst
Insertion sort	$\Theta(n)$	$\Theta(n^2)$	$\Theta(n^2)$
Merge sort	$\Theta(n \log n)$	$\Theta(n \log n)$	$\Theta(n \log n)$
Quick sort	$\Theta(n \log n)$	$\Theta(n \log n)$	$\Theta(n^2)$

4.

Valet av sorteringsalgoritm beror på vilken datastruktur som ska sorteras, storleken på datamängden, om man är beroende av stabil sortering eller inte, och så vidare. Insertion sort kan fungera bra om man till exempel vet att man bara kommer sortera väldigt små datamängder.

5.

Se föreläsningsslides.