

# SI-möte #13, Programkonstruktion och datastrukturer

Elias Castegren & Kristiina Ausmees

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

21 februari 2012

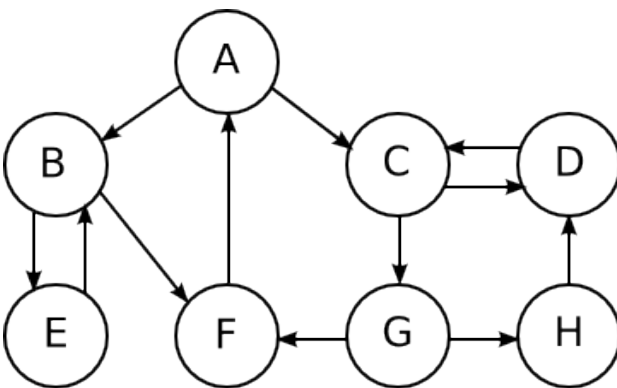
## Begrepp

- i) Vad är en graf? Förklara orden nod (*node* eller *vertex*) och kant (*edge*).
- ii) Vad menas med en nods grad (*degree*)?
- iii) Vad är en väg (*path*) i en graf?
- iv) Vad är skillnaden på en riktad och en oriktad graf?
- v) Är träd också grafer? När är en graf inte ett träd?
- vi) Vad är en komponent (strongly connected component) i en graf?

## Övning

1.

Betrakta nedanstående graf:



- i) I vilken ordning besöks noderna i en Breadth-First-Search (BFS)?
- ii) I vilken ordning besöks noderna i en Depth-First-Search (DFS)?
- iii) Är grafen ett träd?
- iv) Är grafen en komponent?
- v) Finns det någon väg som besöker alla noder exakt en gång?

## Fler begrepp

- i) Hur fungerar Insertion sort? Beskriv algoritmen med ord, inte med kod. Vad är det bästa, respektive sämsta utgångsläget för Insertion sort?
- ii) Hur fungerar Merge sort? Vilka är de tre stegen i algoritmen? Vad är basfallet i rekursionen?
- iii) Hur fungerar Quick sort? Beskriv algoritmen med ord, inte med kod. Hur går partitioneringen till i bästa respektive sämsta fall?
- iv) Vad menas med att en sorteringsalgoritm är stabil?

## Övningar

### 2.

Sortera följande listor i stigande ordning för hand (på papper eller med spelkort) med Insertion sort, Merge sort (dela listorna rakt av på mitten) och Quick sort (välj det första elementet som pivotelement). Räkna antalet heltalsjämförelser (och antalet delningar med Merge sort) för varje algoritm och försök förutsäga vilken algoritm som kommer ge det minsta antalet operationer (du behöver inte räkna med tiden det tar att flytta på ett element).

i) [1, 3, 2, 5, 6, 4, 8, 7, 9, 10]      ii) [6, 8, 1, 3, 10, 9, 7, 2, 4, 5]

iii) [7, 8, 3, 10, 9, 2, 1, 4, 6, 5]      iv) [10, 9, 8, 6, 7, 5, 4, 3, 1, 2]

Slumpa en ny lista och sortera på valfritt sätt igen om du vill förstå någon av algoritmerna bättre!

### 3.

Fyll i tabellen med tidskomplexiteten för algoritmerna i bästa, genomsnittliga och värsta fall.

|                | Best | Average | Worst |
|----------------|------|---------|-------|
| Insertion sort |      |         |       |
| Merge sort     |      |         |       |
| Quick sort     |      |         |       |

### 4.

När är de olika sorteringsalgoritmerna att föredra? Finns det någon poäng med att använda Insertion sort trots att algoritmen i allmänhet har högre komplexitet?

### 5.

Skriv implementationer av Merge sort och Quick sort i ML-kod (helst utan att titta på koden i föreläsningsnoterna) för heltalslistor. Vilka hjälpfunktioner behöver du? Börja med funktionsspecifikationerna! Skriv sedan upp en rekursion och hitta en sluten form.

---

*Lycka till!*