

# SI-möte #10, Programkonstruktion och Datastrukturer

Elias Castegren & Kristiina Ausmees

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

31 januari 2012

## Begrepp

- i)* Vad är en sidoeffekt? Vad är skillnaden mellan en sidoeffekt och ett eftervillkor?
- ii)* Vad är en ström? Till vad används de speciella strömmarna `stdIn` och `stdOut`?

## Övningar

1.

Skriv en funktion `printList(l) : string list -> unit` som skriver ut alla strängar i `l` på varsin rad i terminalen.

2.

Skriv en funktion `lineNumbers(f) : string -> unit` som skapar en ny fil med namnet `f.numbered`, med samma innehåll som filen `f`, men med radnummer först på varje rad. Raden

*En Gul Banan*

i `f` blir alltså

*1. En Gul Banan*

i `f.numbered` om det är den första raden. Fundera på hur du kan hålla ordning på vilket radnummer funktionen är på.

3.

Skriv en funktion `guessNumber(n)` som låter en användare gissa ett slumpat tal mellan noll och `n`. Om man gissar fel ska funktionen berätta om talet är större eller mindre än gissningen och låta spelaren gissa igen. Funktionen ska terminera om spelaren gissar rätt eller matar in en tom rad. Vid terminering ska det rätta svaret returneras. Antag att det finns en funktion `randomRange(a, b)` som beräknar ett slumpat tal mellan `a` och `b`.

(`Int.fromString(s) : string -> int option`, kan vara användbar)

## Fler begrepp

- i)* Vilka egenskaper har en *heap*? Vad är det karaktäristiska för elementens placering i en min-heap (eng. *heap property*)?
- ii)* Vad är en *binär heap*?
- iii)* Vad är ett *binomialt träd*? Vad menas med rangen av ett binomialt träd? Hur många noder har ett binomialt träd av rang  $k$ ? Hur många noder har trädet på nivå  $i$  och vad har trädet för höjd? Kan ett binomialt träd vara tomt?
- iv)* Vad är en *binomial heap*? Hur är träden i en binomial heap ordnade och vad gäller för trädens rang? Kan en binomial heap vara tom?
- v)* Är en binomial heap ett träd?

### 4.

Lägg in nedanstående element (i given ordning) i en binomial min-heap.

4, 3, 1, 7, 2, 6, 5

Skriv upp hur heapen ser ut i varje steg (det kan underlätta att rita upp heapen även i mellanstegen när träden struktureras om).

### 5.

Ta bort det minsta elementet ur den resulterade heapen i uppgift 1 och se till att heapstrukturen behålls.

### 6.

Ange rangerna hos de binomiala träden i en binomial heap med

- i)* 52 element
- ii)* 53 element
- iii)* 127 element
- iv)* 128 element

Hur ser det värsta fallet ut när man ska lägga till ett element i en binomial heap och kring vilka värden ligger antalet element i heapen då?

### 7.

Beskriv hur man kan använda en heap (binär eller binomial) som “mellanlandning” för elementen när man sorterar en lista. Ge en informell härledning av tidskomplexiteten i det värsta fallet. (Algoritmen kallas *heapsort*)

---

*Lycka till!*