KTH, Nada, Alexander Baltatzis

2D1320, TENTAMEN I TILLÄMPAD DATALOGI
Lördagen den 13 mars 2004 kl 8–13

Maxpoäng = 50. Betygsgränser: 25 poäng ger trea, 35 ger fyra, 45 ger femma. Resultatet anslås senast 13 februari på Nadas anslagstavla. Hjälpmedel : En algoritmbok och det rosa formelbladet.

(5p) **1.** *Labyrinten*

Olle går ensam omkring i en labyrint, hans kompisar har redan hittat ut. Det finns gångar och korsningar. En del gångar är blindgångar. Vid varje korsning kan man gå åt höger eller vänster alternativt gå tillbaka dit man kom ifrån. Olle bestämmer sig för att hålla till höger i varje korsning tills han hittar utgången. Vad kallas en sådan sökalgoritm? Vad behöver man för datastrukturer för att implementera en sådan algoritm?

2. *Kökod*

En implementation av en kö fungerar inte som det var avsett (se nedan). Om man stoppar in "Anna", "Bengt" och "Cecilia" i kön. Rita och förklara hur kön ser ut efter varje instoppning.

```
class Node {
    Node next;
    Object data;
    Node(Object o) {
        data = o;
    }
}
class Queue {
    Node head;

    void put(Object o) {
       head = putInto(o, head);
    }
    Node putInto(Object o, Node p) {
        if (p == null) return new Node(o);
        else return putInto(o, p.next);
    }
}
```

Formulera en rekursiv tanke för att stoppa in ett element i en kö och rätta till metoden putInto så att den fungerar.

3. *Syntax*

Olle sitter och rättar ett tentatal. Tentatalet går ut på att man ska skriva en grammatik för meddelande av följande typ:

```
Båt 42, båt 666, båt 4711 och båt 17 ska in!
Båt 1 och båt 2 ska in!
Båt 13 ska in!
```

Vilken eller vilka av följande alternativ kan producera dessa meddelanden? Motivera varför de övriga inte kan producera dem (ange exempel på meddelanden som inte kan produceras).

En del av alternativen kan producera oönskade meningar, man vill t.ex. inte ha Båt 1 och båt 2, båt 3 och båt 4 ska in!"

Vilken eller vilka av alternativen kan producera oönskade meningar? Ge exempel.

```
<meddelande> ::= Båt <tal><svans> | <meddelande> båt <tal><svans>
<svans>      ::= och | ska in! | ,
<tal>        ::= 1 | 2 | 3 | ...

<meddelande> ::= Båt <tal> ska in! | Båt <tal> <svans>
<svans>      ::= och båt <tal> ska in! | , båt <tal> <svans>
<tal>        ::= 1 | 2 | 3 |...

<meddelande> ::= Båt <tal><svans>
<svans>      ::= ska in! | , båt <tal> | och båt <tal>
<tal>        ::= <svans> | 1 | 2 | 3 |...

<meddelande> ::= Båt <tal><svans> | båt<tal><svans>
<svans>      ::= ska in! | , | och
<tal>        ::= 1 | 2 | 3 | ...
```

4. *Heap*

Komplexiteten för att stoppa in ett element i en heapvektor är O(log N). Beskriv hur man stoppar in ett element i en heapvektor och motivera varför det blir O(log N).

Visa hur en heapvektor ser ut efter varje insättning om man i en tom heapvektor stoppar in elementen 133, 520, 800, 13, 87, 900. Lägst nummer har högst prioritet!

VGV

5. *Simbassängen*

En vattenrutschbana mynnar ut i en trång bassäng med trappan i fel ände. Hinner jag inte upp förrän åkaren efter plumsat i måste alltså denne gå upp före mej. Egentligen får en åkare inte ge sej iväg förrän de tidigare hunnit upp vilket syns med ett rött/grönt trafikljus vid startpositionen. Men en del busungar struntar konsekvent i det och åker mot rött så ibland trängs många i bassängen i väntan på en skötsam åkare.

Om rutschbanekön uppfattas som en abstrakt kö av barn där varje barn har egenskaperna namn och busighet. Vilken abstrakt datastruktur är då bassängen? Beskriv en algoritm som simulerar rutschbanan en förmiddag när alla barn åker hela tiden. Den som står först i kön (ett skötsamt barn som inväntar sin tur) hinner precis åka tio gånger och man vill veta hur många gånger den som står näst först (en busunge som åker mot rött) då åkt.

6. *KMP*

När man vill matcha en sträng mot en annan sträng kan man använda en algoritm av Knuth, Morris och Pratt. Då behöver man tillverka en next-vektor. Vad har man den till?

Vad blir next-vektorn för följande mening? (Du behöver inte rita automaten)

KANADAGÄSS KAN KANSKE KALLAS KANADENSISKA

7. *Dokuåpan*

En populär dokuåpa söker deltagare. Flera (243 personer) vill vara med och arrangörerna sorterar de sökande i en lång rad efter näslängd.

När programledaren öppnar dörren för att släppa in den första sökanden får en handfull av de som står först i kön syn på honom och blir så skraja att de springer och gömmer sig i kön. Beskriv en smart algoritm för att sortera om kön. Antag att det visade sig vara tre personer som blev skrämda, hur många näsjämförelser behövs det med din algortitm för att sortera om kön?

8. *Kollisioner*

Redogör för kollisionshantering i hashtabeller. Måste man alltid hantera krockar? Vad händer om man inte gör det? Vad innebär klustring?

Redogör för kollisionshantering i bloomfilter, hur fungerar det?