

Måndagen den 11 mars 2013 kl 14–18

Maxpoäng = 100. 50 poäng ger E, men den som fått 47-49 poäng kan få komplettera. Gränserna för högre betyg är 60, 70, 80, och 90 poäng. Skriv upp antal bonuspoäng från labbar respektive hemtal på tentaomslaget.

Hjälpmedel: En algoritmbok och ett eller flera handskrivna formelblad.

1. *Ultimat automat*  
(8p) I melodifestivalen i förrgår skilde sig den utländska juryns bedömning från svenska folkets. Juryn sökte kanske efter det ultimata wailandet? Rita en KMP-automat som matchar YOOUIYOOUYOUH.  
Ange även next-vektorn.
  
2. *Ukrainsk paranoia*  
(10p) Den ukrainska juryn är orolig för att deras hemliga jury-resultat avlyssnas och även förvanskas på vägen till Sverige. Den svenska festivalledningen tar hjälp av en skärpt tilda-student som löser båda problemen med ett koncept av publika och privata nycklar. Beskriv vilka problem det är som löses med konceptet, och hur det görs, utan att gå in för mycket på matematiska detaljer.
  
3. *Utlandets prioritering*  
(10p)
  - a. Det ryktas om att melodifestivalledningen ursprungligen hade tänkt slänga ut de fem bidrag som fick minst röster av den utländska juryn genom att använda en min-heap. Visa hur prioritetskön nedan ser ut efter varje bortplockning när man plockar ut de fem sämsta bidragen.  
[8, 30, 33, 49, 37, 57, 36, 82, 91, 50]
  
  - b. De svenska rösterna ska läggas i en min-heap så att programledaren kan börja läsa upp det bidrag som fått minst röster. Fem bidrag är redan insatta i prioritetskön  
[32, 40, 41, 75, 103]

Visa hur kön ser ut efter varje insättning av resterande poäng  
44, 12, 18, 59, 49

(18p) 4. *Vann rätt låt?*

Aftonpressen har frågat runt och fått 1000 osorterade svar (ja/nej) från 1000 orter.

```
Alingsås      0 0 0 1 1 0 1 1 1 0 1 0 1 1...
Alvesta       1 1 1 1 1 1 0 1 1 0 1 1 1 0...
Arboga        0 0 1 0 0 0 0 1 0 0 0 0 1 0...
.
.
.
```

Det skrivs pythonprogram för att ta reda på vilken ort som har flest ja-röster (ettor, nej-röster är nollor).

- a. Vilka av följande kodexempel returnerar antal ja-röster i den eller de orter som varit mest positiva till vinnarlåten (raden med flest ettor med andra ord).

```
# counts 1 in each row in matrix m
def foo1(m):
    max_ones = -1
    for row in m:
        ones = 0
        for x in row:
            ones += x
        if ones > max_ones:
            max_ones = ones
    return max_ones
```

```
# counts 0 in each matrix m
def foo2(m):
    nr_of_bits = len(m[0])
    min_zeros = nr_of_bits
    for row in m:
        zeros = 0
        for x in row:
            if x == 0:
                zeros += 1
        if zeros < min_zeros:
            min_zeros = zeros
    return nr_of_bits - min_zeros
```

```

# same as foo2 but with an extra if
def foo3(m):
    nr_of_bits = len(m[0])
    min_zeros = nr_of_bits
    for row in m:
        zeros = 0
        for x in row:          # inner for-loop
            if x == 0:
                zeros += 1
                if zeros >= min_zeros:
                    break      # break out of inner for-loop
        if zeros < min_zeros:
            min_zeros = zeros
    return nr_of_bits - min_zeros

# first 10 % as foo2 rest as foo3
def foo4(m):
    nr_of_bits = len(m[0])
    first_rows = len(m) // 10    # integer division, first 10 % rows
    m1 = m[0 : first_rows]
    max_ones = foo2(m1)          # call foo2
    min_zeros = nr_of_bits - max_ones
    m2 = m[first_rows : len(m)]
    for row in m2:
        zeros = 0
        for x in row:
            if x == 0:
                zeros += 1
                if zeros >= min_zeros:
                    break
        if zeros < min_zeros:
            min_zeros = zeros
    return nr_of_bits - min_zeros

```

- b. Är det någon av de fungerande algoritmerna som är effektivare än de andra? Vilken eller vilka och under vilka förutsättningar?
- c. Vilken komplexitet har de fungerande algoritmerna som funktion av antal rader och kolumner? Motivera kortfattat.

5. *Utred bloomfilter*

- (8p) Det sägs att det här med bloomfilter är bra när man har många ettor och nollor. Utred huruvida man kan använda bloomfilter för att jämföra om två matriser med ettor och nollor är lika.

6. *Uppfinn sorteringsalgoritm*

(14p) Den här matrisen med ja- och nej-röster (ettor och nollor) som aftonpressen sammanställde skulle man vilja sortera så att ettorna (ja-rösterna) står för sig på varje rad. Beskriv en effektiv metod för att sortera matrisen. Ange komplexitet för din algoritm. Exakt hur många jämförelser blir det med din algoritm givet en  $1000 \times 1000$  matris?

7. *Utmärkta algoritmer och datastrukturer*

- (20p)
- a. Anta att vi nu har en sorterad  $1000 \times 1000$  matris med ettor och nollor. Beskriv en algoritm som returnerar både vilken rad (eller vilka rader om flera har samma) som har flest ettor samt hur många ettor det är på den raden.
  - b. Vilken komplexitet har din algoritm?
  - c. Hur många jämförelser blir det med din algoritm i en  $5 \times 10$  matris med 2, 4, 7, 5 samt 3 ettor.
  - d. Beskriv de datastrukturer som din algoritm tar som indata, och vad för slags utdata den returnerar.

(12p) 8. *Uppdrag: skriv syntax*

- a. Ovanstående algoritmer fungerar troligen dåligt om folk velar och svarar annat än ja eller nej (1 eller 0). Skriv en syntax som känner igen binära tal.
- b. Skriv en syntax som känner igen binära tal som är jämnt delbara med 8.