Laboration 1, uppgift 2 i TDA416

Grupp 7: Erik Öhrn, Paula Eriksson Imable

2015-01-30

1 Komplexitetsanalyser

1.1 Handvändning:

I det värsta fallet körs den inre loopen n gånger.

1. Det är tre nästlade for-loopar. Detta ger
$$T(n) \in O(n^3)$$
2. Det är två nästlade for-loopar, vilket då ger
$$T(n) \in O(n^2)$$
3. Det är endast en for-loop, vilket ger
$$T(n) \in O(n)$$

1.2 Matematisk korrekt:

Vi räknar med avseende på aritmetiska operationer, inklusive =. Se bilagor för uträkning. Angående for-loopar: de räknas som 1 (initialeringen i början) + summan av de inneslutna operationerna + 1 (jämförelsen körs en sista gång). Math.com[1] användes för att få en formel för $\sum_{i=0}^{n} i^2$.

1. Formeln för algoritm 1.
$$T(n) = \frac{5}{6}n^3 - \frac{5}{2}n^2 + \frac{5}{12}n + 3 \Rightarrow \tag{1}$$
$$T(n) \in O(n^3)$$
2. Formeln för algoritm 2.
$$T(n) = \frac{13}{2}n^2 + \frac{17}{2}n + 3 \Rightarrow \tag{2}$$
$$T(n) \in O(n^2)$$
3. Formeln för algoritm 3. I värsta fall kollar den först if-satsen, finner den falsk men går sedan in i else-if-satsen.
$$T(n) = 10n + 5 \Rightarrow \tag{3}$$
$$T(n) \in O(n)$$

[1] http://www.math.com/tables/expansion/power.htm

1.3 Pedantisk

Vi har valt att analysera algoritm 3:

```
1  public static int maxSubSum3( int[] a ) {
2    int maxSum = 0; // 1 op
3    int thisSum = 0; // 1op
4    for( int i = 0, // 1op
5      j = 0; //1op
6      j < a.length; // (n+1)
7      j++ ) { //2n
8      thisSum += a[j]; //3n -> 2op + fältuppslag varje varv
9      if( thisSum > maxSum ) { //n
10       maxSum = thisSum; //n
11       seqStart = i; // n
12       seqEnd = j; //n
13     }else if( thisSum < 0 ) { //n
14       i = j + 1; // 2n
15       thisSum = 0; //n
16     }
17   } return maxSum;
18 }
```

Vilket i operationer per kodrad ger:

```
1  -
2  1
3  1
4  1
5  1
6  n+1
7  2n
8  3n (2n om inte fältuppslag räknas)
9  n
10 n
11 n
12 n
13 n
14 2n
15 n
16 -
17 -
18 -
```

$= 10n + 4$

I värsta fall körs inte if-satsen men däremot else-if-satsen. Detta ger resultatet ovan. Om fältuppslag räknas med ger det dock samma resultat om if-satsen körs.

Detta ger samma resultat som den matematiskt korrekta analysen.

2 Grafer

2.1 Uppmätt

För grafen anropade vi funktionen 1000 gånger per fältstorlek. Som gränser för storlekarna satte vi för algoritmerna

1. 1024
2. 8192
3. 32768

Som svar erhölls då:

Fältstorlek	Algo 1	Algo 2	Algo 3
64	0,166	0,02	0,005
128	0,806	0,024	0,001
256	5,712	0,1	0,001
512	58,949	0,343	0,002
1024	348,989	1,303	0,005
2048		5,175	0,009
4096		20,581	0,0019
8192		80,252	0,0038
16384			0,0077
32768			0,166

Table 1: Tid i sekunder för olika algoritmer vid olika fältstorlekar, per tusen körningar

Detta ger följande graf:

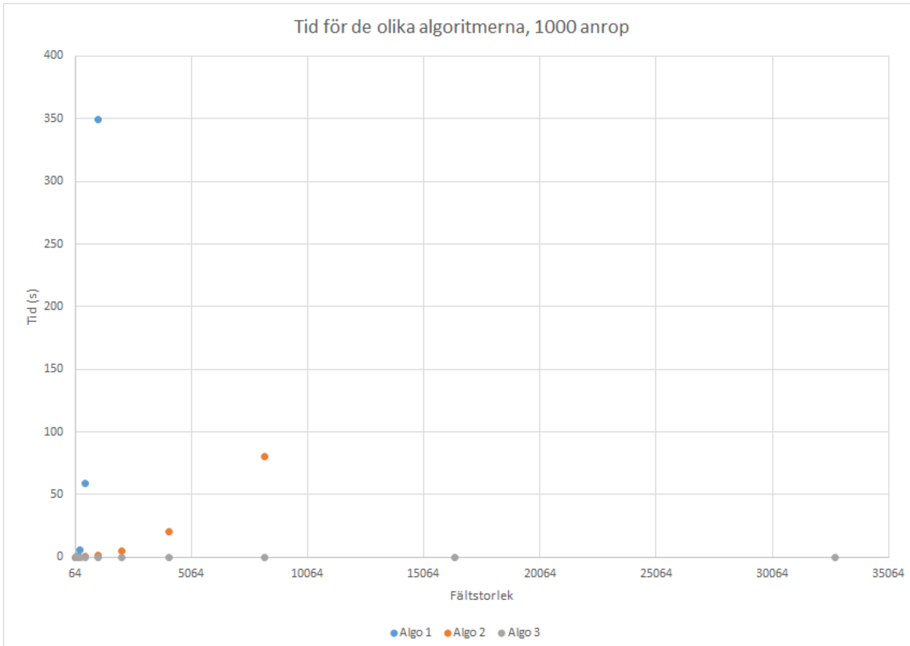

Figure 1: Tid för algoritmkörning

Om grafens x-axel visas som en log_2-skala, samt om linjer läggs till, syns det mer relevant hur de skiljer sig från varandra (i och med det intressanta är hur de skiljer sig när fältstorleken fördubblas).

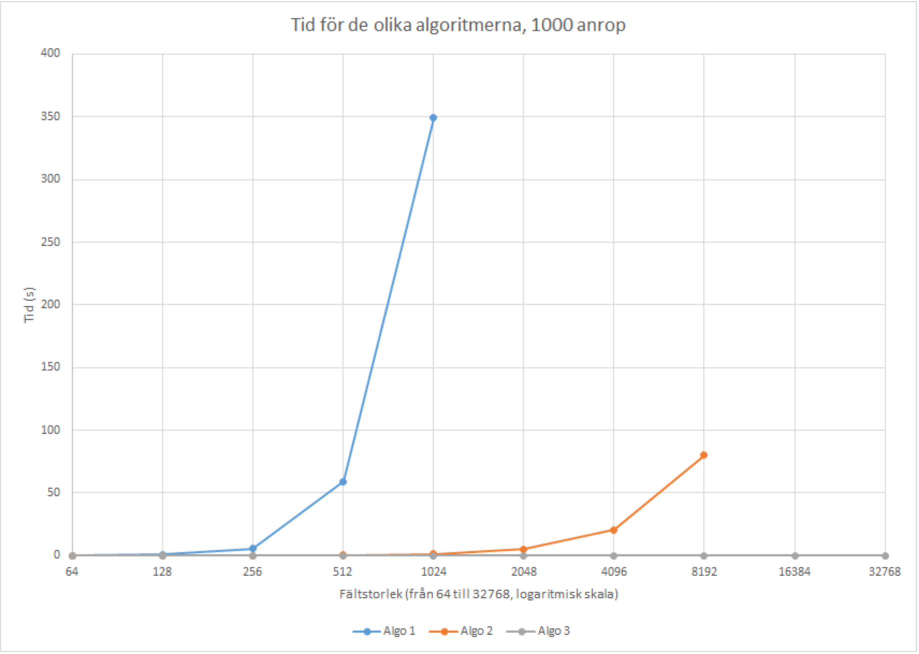

Figure 2: Tid för algoritmkörning, logaritmerad skala

2.2 Beräknat

Detta går att jämföra med beräknat antal operationer.

Fältstorlek	Algo 1	Algo 2	Algo 3
64	208243	27171	645
128	1706723	107587	1285
256	13817283	428163	2565
512	111192963	1708291	5125
1024	892163843	6824451	10245
2048		27280387	20485
4096		109086723	40965
8192		436277251	81925
16384			163845
32768			327685

Table 2: Beräknade antal operationer för olika fältstorlekar

Vilket ger följande grafer:

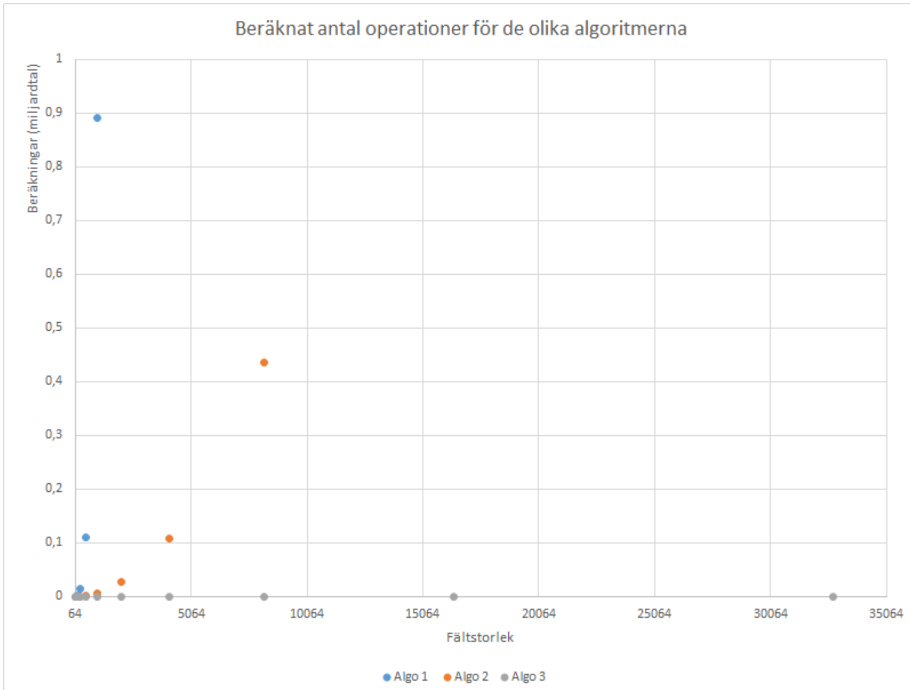

Figure 3: Beräknat antal operationer

samt

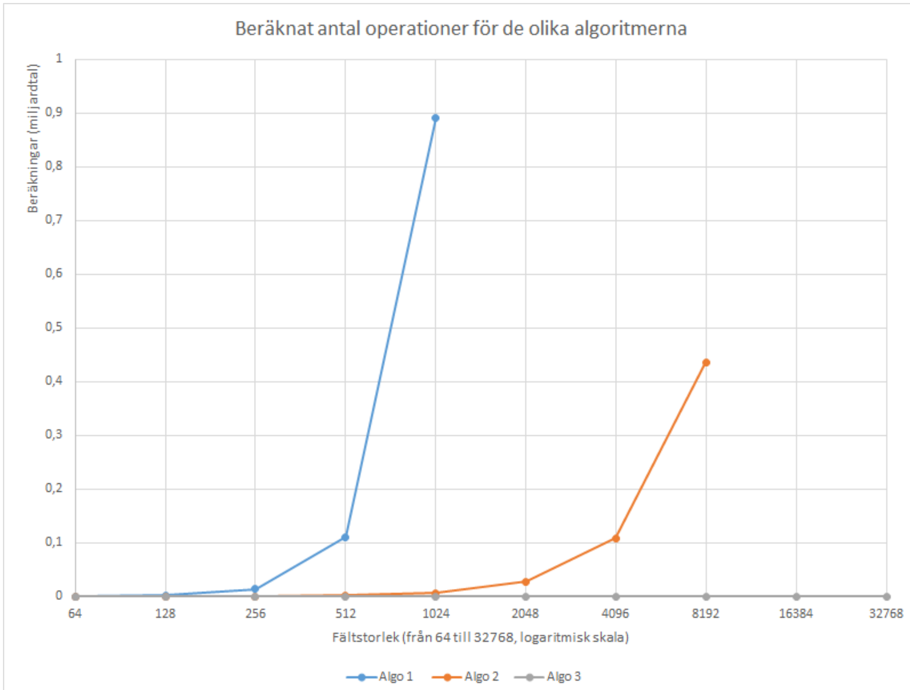

Figure 4: Beräknat antal operationer, logaritmerad skala

2.3 Jämförelse

Algoritm 1 och 3 ser till formen ut att stämma. Algoritm 2 avviker något från de övriga. Vi har uppfattningen av att det beror på if-satsen i den innersta loopen. Vi har räknat med att den alltid visar sant, vilket simulerar ett värsta fall. När den körts har jämförelsen i if-satsen dock varit falsk majoriteten av gångerna. Formeln för algoritm 2 om if-satsen aldrig skulle köras är

$$T(n) = 3n^2 + \frac{17}{2}n + 3 \qquad (4)$$

Vilket ger

Fältstorlek	Algo 2
64	27171
128	107587
256	428163
512	1708291
1024	6824451
2048	27280387
4096	109086723
8192	436277251
16384	
32768	

Table 3: Värden för algoritm 2 vid falsk resultat vid jämförelse

Detta ger grafen:

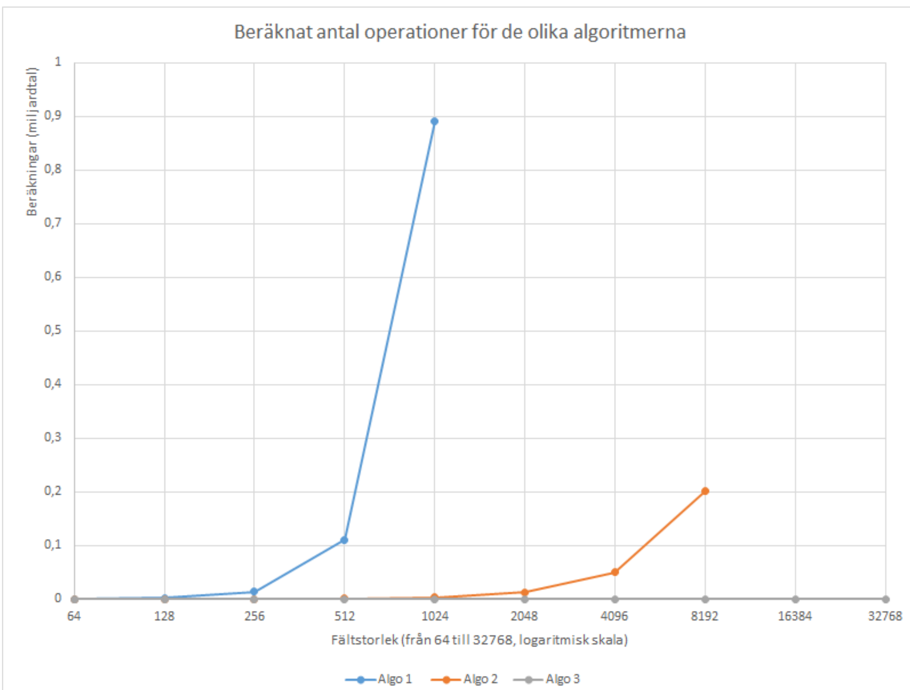

Figure 5: Graf över beräknade operationer, bortsett från if-satsen i algoritm 2, logaritmerad skala

Denna graf liknar den för de uppmätta värdena.

Bilagor

$$1+1+\sum_{i=0}^{n-1}\left(1+2+1+\sum_{j=i}^{n-1}\left(1+2+1+1+\sum_{k=i}^{j}(1+2+2)+1+4\right)+1\right)+1 =$$

$$k_1+\sum_{i=0}^{n-1}k_2+\sum_{i=0}^{n-1}\sum_{j=i}^{n-1}k_3+\sum_{i=0}^{n-1}\sum_{j=i}^{n-1}\sum_{k=i}^{j}5 = k_1+k_2\sum_{i=0}^{n-1}1+k_3\sum_{i=0}^{n-1}\sum_{j=i}^{n-1}1+5\sum_{i=0}^{n-1}\sum_{j=i}^{n-1}\sum_{k=i}^{j}1$$

$$= k_1+k_2\sum_{i=1}^{n}1+k_3\sum_{i=0}^{n-1}\sum_{j=1}^{n-i}1+5\sum_{i=0}^{n-1}\sum_{j=i}^{n-1}\sum_{k=1}^{j-i+1}1 = k_1+k_2n+k_3\sum_{i=0}^{n-1}(n-i)+5\sum_{i=0}^{n-1}\sum_{j=i}^{n-1}(j-i+1)$$

$$= k_1+k_2n+k_3\sum_{i=1}^{n}(n-i-1)+5\sum_{i=0}^{n-1}\sum_{j=1}^{n-i}(j-i+1+i-1) =$$

$$= k_1+k_2n+k_3n\sum_{i=1}^{n} -k_3\sum_{i=1}^{n}i-k_3\sum_{i=1}^{n}1+5\sum_{i=0}^{n-1}\sum_{j=1}^{n-i}j =$$

$$k_1+k_2n+k_3n^2-k_3\frac{n(n+1)}{2}-k_3n+5\sum_{i=0}^{n-1}\frac{(n-i)(n-i-1)}{2}$$

$$= k_1+k_5n+k_6n^2+\frac{5}{2}\sum_{i=0}^{n-1}(n^2-n-2ni+i^2+i) = k_1+k_5n+k_6n^2+\frac{5}{2}\sum_{i=1}^{n}(n^2-n-2n(i-1)+(i-1)^2+(i-1)$$

$$= k_1+k_8n+k_7n^2+\frac{5}{2}n^3-5n\sum_{i=1}^{n}i+\frac{5}{2}\sum_{i=1}^{n}i^2$$

//Summorna för värden som ej kommer ge n^3-termer räknas ej ut, men k-värdena uppdateras istället.

$$= k_1+k_{10}n+k_9n^2+\frac{5}{2}n^3-5n\frac{n(n+1)}{2}+\frac{5}{2}\cdot\frac{1}{3}n^3$$

$$= k_1+k_{10}n+k_{11}n^2+\frac{5}{2}n^3-\frac{5}{2}n^3+\frac{5}{6}n^3$$

$$= \frac{5}{6}n^3+k_{11}n^2+k_{10}n+k_1$$

$$\Rightarrow T(n)\in O(n^3)$$

$k_1 = 3$

$k_2 = 5$

$k_3 = 10$

$k_4 = N/A$

$k_5 = k_2-\frac{k_3}{2} = 5-\frac{10}{2} = 0$

$k_6 = k_3-\frac{k_3}{2} = 10-\frac{10}{2} = 5$

$k_7 = k_6-\frac{5}{2}\cdot\frac{15}{4} = -\frac{5}{4}$

$k_8 = k_5 = 0$

$k_9 = k_7+\frac{5}{2}\cdot\frac{1}{2} = -\frac{5}{4}+\frac{5}{4} = 0$

$k_{10} = k_8+\frac{1}{6}\cdot\frac{5}{2} = \frac{5}{12}$

$k_{11} = k_9-\frac{5}{2} = -\frac{5}{2}$

$$\Rightarrow T(n) = \frac{5}{6}n^3-\frac{5}{2}n^2+\frac{5}{12}n+3$$

$$1+1+\sum_{i=0}^{n-1}\left(1+2+1+1+\sum_{j=i}^{n-1}(1+2+2+4)+1\right)+1 = k_1 + k_2\sum_{i=0}^{n-1}1 + 9\sum_{i=0}^{n-1}\sum_{j=i}^{n-1}1 =$$

$$k_1 + k_2 \cdot n + 9\sum_{i=0}^{n-1}(n-i) = k_1 + k_2 n + 9n\sum_{i=0}^{n-1}1 - 9\sum_{i=0}^{n-1}i = k_1 + k_2 n + 9n^2 - 9\frac{n(n-1)}{2}$$

$$= k_1 + k_3 n + 9n^2 - 9\frac{n^2}{2} = k_1 + k_3 n + \frac{13}{2}n^2 = T(n)$$

$$\Rightarrow T(n) \in O(n^2) .$$

$$k_1 = 3$$

$$k_2 = 6$$

$$k_3 = k_2 + \frac{9}{2} = \frac{17}{2}$$

$$\Rightarrow T(n) = \frac{13}{2}n^2 + \frac{17}{2}n + 3$$

Om ej if-sats: $3n^2 + \frac{17}{2}n + 3$

Figure 7: Matematisk uträkning för algoritm 2

$$1+1+2+\sum_{j=0}^{n-1}(1+2+2+5)+1 = k_1 + 10\sum_{j=0}^{n-1} 1 = k_1 + 10n$$

$$\Rightarrow T(n) \in O(n)$$

$$k_1 = 5$$

$$\Rightarrow T(n) = 10n + 5$$

Figure 8: Matematisk uträkning för algoritm 3