Laboration 1 TDA416

Uppgift 3: Komplexitetsberäkning

a)

Algoritmen “MaxSum” räknar ut alla sammanhängande sekvenser av ett fält med tal och returnerar den största sekvensen. Den sparar även start och slutvärden för den maximala sekvensen.

b)

Handviftningsberäkning av första algoritmen

- Den yttersta loopen, betecknad med i, går maximalt *n* varv, där *n* betecknar längden på fältet.
- Den mittersta loopen, betecknad med j, går maximalt *n* varv, där *n* betecknar längden på fältet.
- Den innersta loopen, betecknad med k, går maximalt *n* varv den med, där *n* betecknar längden på fältet.

Detta ger att det maximala antalet totala varv i någon loop blir:

$$T(n) = n * n * n = n^3 \ \in O(n^3)$$

Handviftningsberäkning av andra algoritmen

- Den yttre loopen, betecknad med i, går maximalt *n* varv, där *n* betecknar längden på fältet.
- Den inre loopen, betecknad med j, går maximalt *n* varv, där *n* betecknar längden på fältet.

Detta ger att det maximala antalet totala varv i någon loop blir:

$$T(n) = n * n = n^2 \ \in O(n^2)$$

Handviftningsberäkning av tredje algoritmen

- Loopen, betecknad med i, går maximalt *n* varv, där *n* betecknar längden på fältet.

Detta ger att det maximala antalet totala varv i någon loop blir:

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

Matematisk korrekt uppskattning av första algoritmen

$$\sum_{i=0}^{n-1}\sum_{j=i}^{n-1}\sum_{k=i}^{j} = \sum_{i=0}^{n-1}\sum_{j=i}^{n-1}(j-i+1) = \sum_{i=0}^{n-1}(\sum_{j=i}^{n-1} j + \sum_{j=i}^{n-1}(1-i)) =$$
$$= \sum_{i=0}^{n-1}(-\tfrac{1}{2}(i-n)(i+n-1) + (i-1)(i-n) =$$
$$= \tfrac{1}{2}\sum_{i=0}^{n-1}(-(i^2+i-n+n^2) + 2(i^2-in-i-n)) = \tfrac{1}{2}\sum_{i=0}^{n-1}(i^2-i-2in\ +n^2+n) =$$
$$= \tfrac{1}{2}\sum_{i=0}^{n-1}(i^2-2in-i) + \sum_{i=0}^{n-1}(n^2+n) = \tfrac{1}{2}(-\tfrac{2}{3}(n-1)(n^2+n)) + n^3+n^2) =$$
$$= \ldots = -\tfrac{1}{3}(n^3-n) + \tfrac{1}{2}(n^3+n^2) = \tfrac{1}{2}n^3 - \tfrac{1}{3}n^3 + \tfrac{1}{2}n^2 + \tfrac{1}{3}n = \tfrac{1}{6}(n^3+3n^2-2n)$$

Matematisk korrekt uppskattning av andra algoritmen

$$\sum_{i=0}^{n-1}\sum_{j=i}^{n-1} 1 = \sum_{i=0}^{n-1}((n-1)-1+1) = \sum_{i=0}^{n-1}(n-i) = \sum_{i=0}^{n-1} n - \sum_{i=0}^{n-1} i = n^2 - \tfrac{1}{2}(n-1)n =$$
$$= n^2 - \tfrac{1}{2}n^2 - \tfrac{1}{2}n = \tfrac{1}{2}(n^2-n)$$

Matematisk korrekt uppskattning av tredje algoritmen

$$\sum_{i=0}^{n-1} 1 = n \ \in O(n)$$

Pedantisk analys av tredje algoritmen

Radnummer:	Antal Operationer:
`int maxSum = 0;`	1op.
`int thisSum = 0;`	1op.
`for( int i = 0;`	1op.
`j = 0;`	1op.
`j < a.length;`	1op. * (n + 1)
`j++ ) {`	2op. * n
`thisSum += a[ j ];`	3op. * n
`if( thisSum > maxSum ) {`	3op. * n (2 läs, 1 skriv)
`maxSum = thisSum;`	2op. * n
`seqStart = i;`	2op. * n
`seqEnd = j;`	2op. * n
`}`	
`else if( thisSum < 0 ) {`	2op * n // Sker ej i värsta fallet.
`i = j + 1;`	3op * n // Det är lika dyrt som första valet.
`thisSum = 0;`	1op * n // Så vilket vi väljer spelar ingen roll.
`}`	
`}`	
`return maxSum;`	1op.

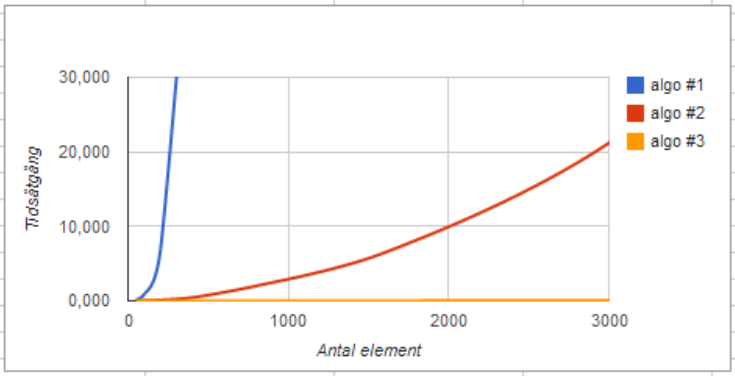
Vilket ger:

$$T(n) = 1+1+1+1+1+(n+1)+((2+3+3+2+2+2)*n)+1 =$$
$$= 16n + 7 \in O(n)$$

c)

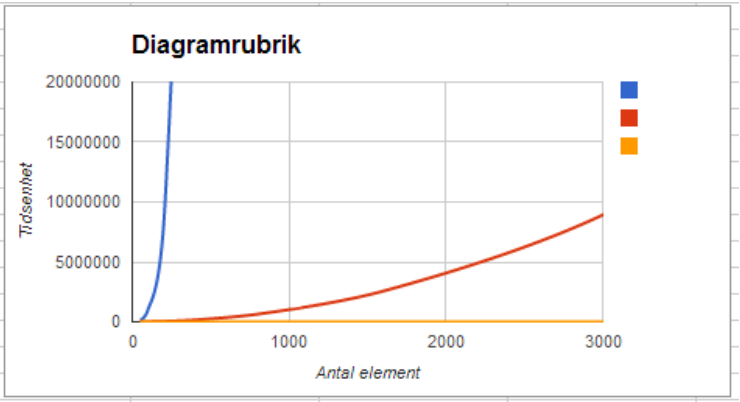
Resultat från testprogram:

Size of array	algo #1	algo #2	algo #3
50	0,134	0,010	0,003
100	0,857	0,027	0,001
200	6,410	0,108	0,002
400	54,935	0,435	0,005
800		2,002	0,009
1600		6,461	0,023
3200		24,296	0,035
6400		96,826	0,072
12800			0,143
25600			0,302
51200			0,601
102400			1,206
204800			2,552
409600			5,390



Resultat av matematisk uppskattning:

Size of array	O(n^3)	O(n^2)	O(n)
50	125000	2500	50
100	1000000	10000	100
200	8000000	40000	200
400	64000000	160000	400
800		640000	800
1600		2560000	1600
3200		10240000	3200
6400		40960000	6400
12800			12800
25600			25600
51200			51200
102400			102400
204800			204800
409600			409600



Vi förväntar oss att vårt testdata förhåller sig väldigt likt de framräknade resultaten. Det ser vi också när vi sedan jämför våra resultat jämfört med de förväntade. Vilket betyder att verkligheten faktiskt stämmer bra överens med teorin.

- Algoritm 1 får en kubisk graf
- Algoritm 2 får en kvadratisk graf
- Algoritm 3 får en linjär graf.

Algoritm 3 växer väldigt fort för världen större än 100000, vilket är oväntat. Antagligen beror detta på någon fysik begränsning i datorn, gissningsvis långsam minneshantering.