## Redovisning 1

for(int i=1;i<=n;i++)

for(int j=1;j<=i;j++)

for(int k=j;k<=i+j;k++)

for(int m=1;m<=i + j- k;m++)

r++;

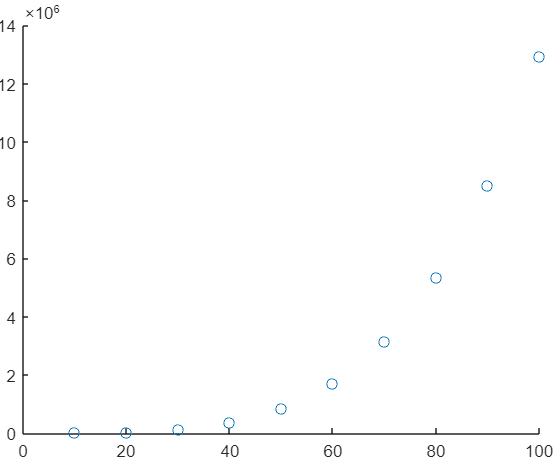
Given sats har körts i Matlab för att undersöka dess tidskomplexitet såväl grafiskt som i tabellformat.  
  
Satsen har körts med olika värden på *n* för att se hur det förhåller sig till tid och antal operationer på r.

Nedan presenteras resultatet för de olika värdena på n och step i en tabell:

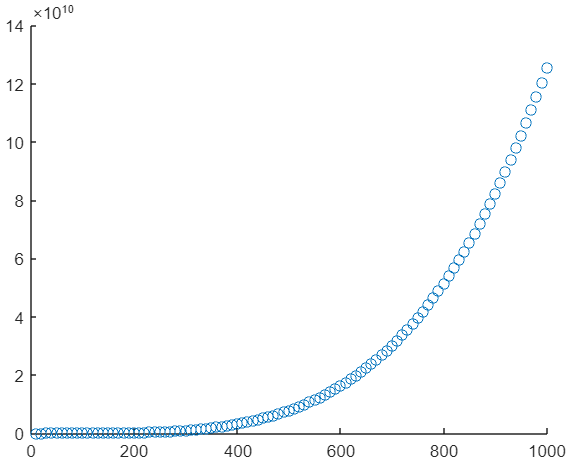
| ***n*** | **tid i mikrosekunder** | **operationer på *r*** |
| --- | --- | --- |
| 10 | 126 | 1 705 |
| 50 | 354 | 834 275 |
| 100 | 985 | 12 920 425 |
| 500 | 23 934 | 7 864 677 125 |
| 1000 | 197 778 | 125 417 041 750 |

Här ses en kraftig exponentiell ökning i tid såväl som operationer på r. Det visar på att satsens tidskomplexitet är dålig. Innan vi går vidare presenterar vi detta i form av en graf.

Här används Matlab för att presentera resultatet i en *scatter plot* (punktdiagram?). Dock beräknas värden både i en kort range *1:1:10* och i en utbredd lång range *10:10:1000* (<start>:<step>:<end>). Grafens koordinater visar (x,y) på (n, r).

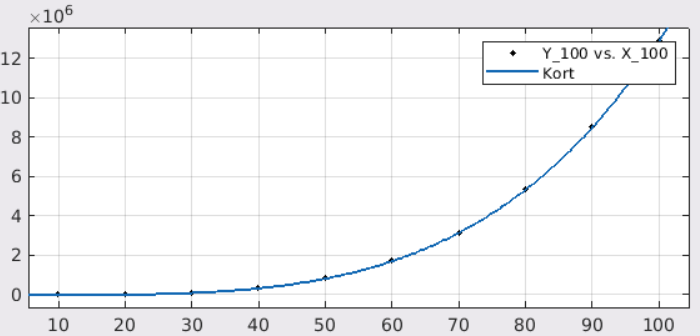
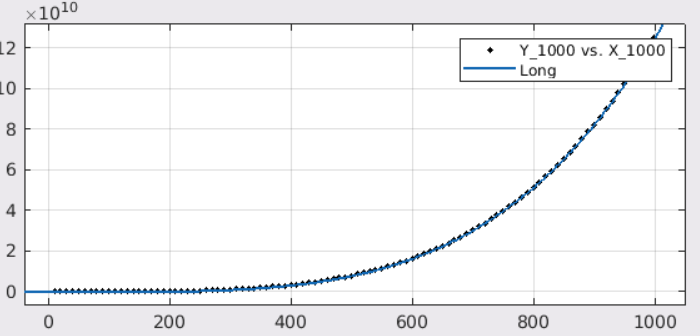


*Figur 1.* En kort range som visar på exponentiell ökning.

  
*Figur 2.* En lång range som visar på samma ökning som i *Figur 1*.

För att ta fram polynomet används Matlabs *Curve fitting*. Polynomet som stämmer överens perfekt med datapunkterna är (samma för både lång och kort range):

**Polynom:**

*****Figur 3.* Ett uppskattat polynom baserat på denna korta rangen  


*Figur 4.* Ett uppskattat polynom baserat på denna långa rangen

Detta polynom svarar på vilken tidskomplexitet satsen hade - titta på den högsta exponenten. i polynomet ger tidskomplexiteten . Polynomet kan även användas till att uppskatta värden på r och tid för *n* större än vad som används i ovanstående tester.   
T.ex. Hur många operationer på *r* görs då *n* är 10 000?

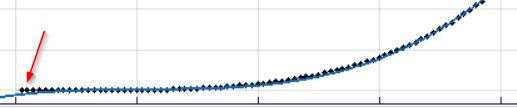
**Slutsats**

Genom att analysera hur satsens körs på olika input-värden (*n*) kunde ett passande polynom tas fram med hjälp av Matlab. Polynomet var av **grad 4** därav kunde slutsatsen dras att satsen i fråga ha tidskomplexiteten .

**Komplettering enligt kommentar:***Hade velat att du tog fram polynom för olika antal punkter och såg att du fick samma polynom, tex n=1:20 och n=1:40. Samt att du anpassade fjärde och femtegradare så att du såg att det inte behövdes högre grad än 4. Vill också att du anger polynomet på exakt bråkform.*

**Analys av olika grad för polynom (***Samt att du anpassade fjärde och femtegradare så att du...***)**

För att visa varför vi kommer fram till polynom av grad 4 testar vi olika grder för att se hur dessa passar polynomet. Vi använder först och främst den större mängden punkter [10:10:1000]. Detta görs med hjälp av matlabs *curve-fitting* verktyg för att hitta polynomet av olika grad, vi kommer fram till följande:

* Grad 3:  
  Genom visuell analys ser vi direkt att polynomet inte täcker alla punkter som skickades in. Det måste vara av högre grad och därmed är det inte en kandidat.  
  
* Grad 4:  
    
  Detta polynom stämmer bra överens med punkterna i vår scatter-plot, därmed en kandidat.
* Grad 5:  
    
  Detta polynom stämmer, likt grad 4, bra överens med punkterna i vår scatter-plot, därmed också en kandidat.

**Grad 4 eller Grad 5? (***Samt att du anpassade fjärde och femtegradare så att du såg att det inte...***)**  
För att avgöra vilken kandidat som gäller för punkterna kollar vi på polynomet Matlab genererade och ser att konstanten framför är 0 och samtliga andra konstanter framför de andra potenserna är samma för både fjärdegradare och femtegradare.   
Detta innebär praktiskt att det är samma polynom. Från denna observation kan vi dra slutsatsen att en fjärdegradare är de lägsta polynomet som stämmer överens med alla punkter.  
Samma resonemang kan givetvis föras vidare för sjättegradre, då kommer dess konstanter framför -potenserna vara 0, och samma konstanter som fjärdegradaren för de andra potenserna.

**Gäller samma för den kortare rangen? (***Hade velat att du tog fram polynom för olika antal...***)**

Vi genererar polynom baserat på den korta rangen [10:10:100]. Vi får då exakt samma polynom: och kan dra samma slutsater som ovan - fjärdegradare eller femtegradare osv.   
  
**Slutsats för polynom: (***Vill också att du anger polynomet på exakt bråkform.***)**  
Framtaget polynom i bråkform blir:  
 som ger oss