

Herons formel

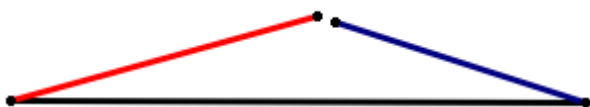
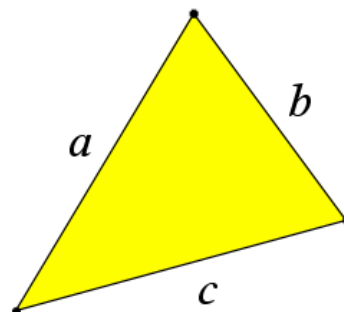
Det här är en aktivitet som inledningsvis handlar om att skriva ett program som beräknar arean av en triangel när man känner till alla sidor. Vi döper sidlängderna till a , b och c . Vi ska också kolla om sidorna överhuvudtaget kan bilda en triangel. Detta undersöks i satsen:

If $a + b > c$ and $a + c > b$ and $b + c > a$

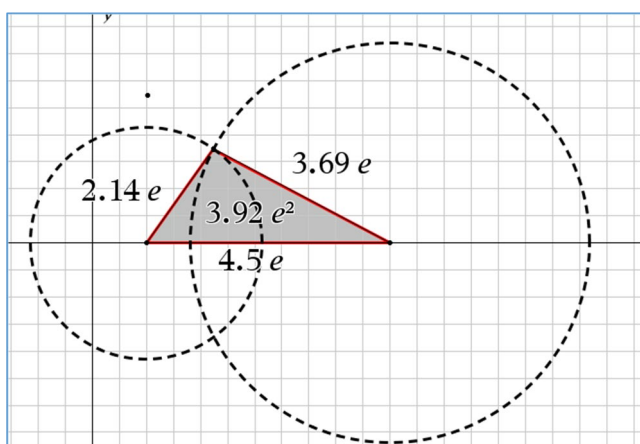
Om de tre villkoren ovan är sanna samtidigt så kan sidorna med längderna a , b och c bilda en triangel. Sidorna med längderna 2, 3 och 6 längdenheter kan inte bilda en triangel. Sammanfattningsvis enligt satsen ovan: $2+3>6$ falskt, $2+6>3$ sant, $3+6>2$ sant.

En av olikheterna är falsk och då kan man inte bilda en triangel.

I figuren nedan ser det ut som den blå och röda linjen tillsammans har mindre längd än den svarta linjen ($b+r<s$)!



Utvidgning: Vi visar nu ett sätt att konstruera en triangel där vi känner de tre sidorna. Figuren är konstruerad med geometriappen i TI-Nspire men du kan naturligtvis göra den med penna, linjal och passare. Vi placerar den längsta sidan efter x -axeln och plottar sedan två cirklar med radier som de två kortare sidorna. Cirklarna har medelpunkter i ändarna på den längsta sidan. Vi ritar sedan de två andra sidorna som har ändpunkter i cirklarnas skärningspunkt. Med TI-Nspire kan man sedan med Herons formel kontrollera att arean verkar korrekt.



Med Herons formel kan du alltså beräkna arean av en godtycklig triangel där du känner längden hos alla tre sidor. Om sidlängderna är a , b och c och s är halva omkretsen, dvs

$$s = \frac{1}{2}(a+b+c)$$

så gäller att

$$\text{Arean} = \sqrt{s(s-a)(s-b)(s-c)}$$

Vi kan pröva genom att mata in längderna för en s.k. egyptisk triangel med sidlängderna 3, 4 och 5. Den är ju rätvinklig och arean är $3 \cdot 4 / 2 = 6$. Halva omkretsen är $(3+4+5)/2=6$. Herons formel ger

$$\text{Arean} = \sqrt{6(6-3)(6-4)(6-5)} = \sqrt{36} = 6$$

Här bredvid har vi nu programmet. Efter inmatning av sidlängderna med tre inputsatser så kontrollerar man att de inmatade sidorna uppfyller villkoren för att de ska kunna bilda en triangel. Det behövs bara en if-sats för detta. Därefter kommer beräkningarna i två steg. Längst ner visar vi en körning av programmet. Vi ser inmatade sidlängder och areaberäkningen på samma skärm.

Pröva också med sidlängder som inte kan bilda en triangel.

```

EDITOR: HERON
PROGRAM LINE 0006
from math import *
a = float(input('mata in längden
             hos sida 1: '))
b = float(input('mata in längden
             hos sida 2: '))
c = float(input('mata in längden
             hos sida 3: '))
#Vi ska nu kolla om de tre längd
erna kan bilda en triangel
if (a+b>c) and (a+c>b) and (b+c>
a):_
# Beräkning av halva omkretsen
....s=(a+b+c)/2
# Beräkning av arean
....arean=sqrt(s*(s-a)*(s-b)*(s-
c))
....print("Arean hos triangeln ä
r ",round(arean,2))
else:
....print("De tre sidorna kan in
te bilda en triangel")
...._

```

```

PYTHON SHELL
>>> # Shell Reinitialized
>>> # Running HERON
>>> from HERON import *
mata in längden hos sida 1: 4
mata in längden hos sida 2: 6
mata in längden hos sida 3: 7
Arean hos triangeln är 11.98
>>> |

```

Fördjupning: Vilka villkor gäller för att tre längder ska kunna bilda en triangel?

Tänk dig att du slumpmässigt viker ett sugrör på två ställen och sedan ser om det går att forma en triangel med det vikta sugröret. För enkelhetens skull antar vi att sugröret är 10 cm långt. Vi ska nu simulera vikningen av sugröret genom att alstra par av slumpstal som representerar var på sugröret vikningen sker.

Vi ska arbeta i statistikeditorn på din grafräknare och vi ska göra 2 st simuleringar i listorna L1 och L2. formeln vi använder är

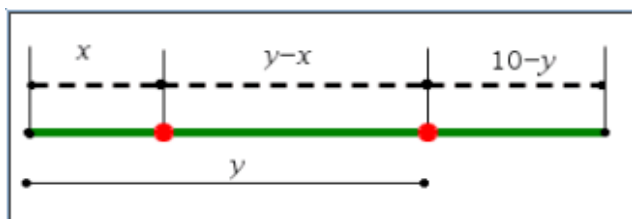
10*slump(100)

Använder du engelsk språkställning heter det *rand* istället för slump.

Vi får då på 100 rader i L1 och L2 slumpstal mellan 0 och 10.

Nu gäller det att för varje par av slumpstal se till att bestämma vilket som motsvarar den första respektive andra vikningen.

Med formlerna " $L3=\min(L1,L2)$ " resp. " $L4=\max(L1, L2)$ " får vi det mindre värdet i lista L3 och det större värdet i lista L4.



Värdena i lista L3 motsvarar då sträckan x i figuren ovan.

I lista L5 beräknar vi mittenlängden som differensen mellan värdena i listorna L1 och L2. Motsvarar längden $y-x$ i figuren.

Vi beräknar vi sedan den sista längden. Motsvarar $10 - y$ i figuren.

Vi har då genom slumpstalsalstring och lite listiga beräkningar värden på de tre längderna. Dessa värden finns alltså i listorna L3, L5 och L6.

Nu kommer det intressanta. I lista VILLK skriver vi

" $L3+L5>L6$ och $L3+L6>L5$ och $L5+L6>L3$ "

Om alla tre villkoren i formeln ovan är sanna så får vi värdet 1 i lista VILLK. Är något eller några av villkoren falska så blir det en nolla

Genom att sätta citattecken runt formlerna i listorna L3 till L6 så uppdateras dessa listor om vi slumpar fram nya värden i L1 och L2.

Andelen "ettor" i lista SANN (står för SANNOLIKHET) beräknas med formeln $\text{sum}(VILLK/100)$. Det blir då ett mått på sannolikheten.

Om du göra många simuleringar bör du få ett värde ung 0,25. Den teoretiska sannolikheten är nämligen $\frac{1}{4}$. På nästa sida finns ett TI-Basic program som gör ett antal simuleringar.

NORMAL FLYT AUTO REELL RAD MP					
L1	L2	L3	L4	L5	S
7.063	4.9471	4.9471	7.063	2.1159	
2.0829	7.5442	2.0829	7.5442	5.4613	
9.0037	2.3562	2.3562	9.0037	6.6475	
4.5195	7.8472	4.5195	7.8472	3.3277	
0.7529	4.4878	0.7529	4.4878	3.7348	
2.0961	3.1956	2.0961	3.1956	1.0995	
9.1087	0.5335	0.5335	9.1087	8.5752	
3.7905	3.937	3.7905	3.937	0.1465	
3.8796	0.0484	0.0484	3.8796	3.8312	
1.5128	5.8821	1.5128	5.8821	4.3693	
5.3226	5.3977	5.3226	5.3977	0.0751	

L5= "L4-L3"

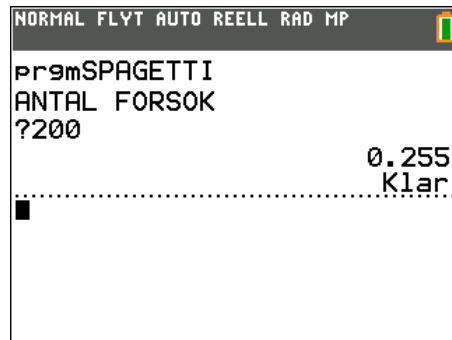
NORMAL FLYT AUTO REELL RAD MP					
L4	L5	L6	VILLK	SANN	S
7.063	2.1159	2.937	1		
7.5442	5.4613	2.4558	0		
9.0037	6.6475	0.9963	0		
7.8472	3.3277	2.1528	1		
4.4878	3.7348	5.5122	0		
3.1956	1.0995	6.8044	0		
9.1087	8.5752	0.8913	0		
3.937	0.1465	6.063	0		
3.8796	3.8312	6.1204	0		
5.8821	4.3693	4.1179	1		
5.3977	0.0751	4.6023	0		

SANN(1)=sum(LVILLK/100)

Här är programmet. Istället för att visa programmet i räknarfönstret har vi här öppnat det i programvaran [TI Connect™ CE](#). Till höger har vi resultatet av en körningen.

```
0→S
0→T
Disp "ANTAL FORSOK"
Input N
For(A,1,N)
Lbl 1
10slump→X
10slump→Y
If X>Y
Goto 1
S+1→S
If X<5 och Y>5 och Y<X+5
T+1→T
End
Disp T/S
```

Utmaning: Försök nu att skriva ett program i Python som gör samma sak.



```
NORMAL FLYT AUTO REELL RAD MP
ProgSPAGETTI
ANTAL FORSOK
?200
0.255
Klar.
```