

OPTIMERING MED PROGRAMMERING

Den här modulen handlar om ekonomiska modeller vid handel och ett skoföretag som ska lansera en ny skomodell. Under lektionen kommer eleverna att få vara rådgivare till skoföretaget och hjälpa dem genomföra olika ekonomiska analyser med hjälp av optimering. De ekonomiska modeller som används är förenklade och något justerade för att passa lektionens målgrupp. Eleverna kommer hjälpa företaget med prissättningen genom att undersöka maximal vinst, kolla på hur många budbilar företaget behöver för att vara lönsamt och hur stor efterfrågan blir vid ett visst pris. För att undersöka de ekonomiska problemen används både matematik och programmering.

Lektionen kommer att kretsa kring ett antal frågeställningar som eleverna ska arbeta med. Meningen med lektionen är att eleverna ska fundera långa stunder själva, att de ska få chansen att utvecklas genom att inte fatta och träna på sin uthållighet. Det gör inget om inte alla problem hinns med, det viktiga är att eleverna får tid att fundera själva. Lektionen är skapad i samarbete med Handelsrådet och det finns en tillhörande powerpointpresentation. Varje kapitel inleds med en introduktion om det ekonomiska samband som kapitlet handlar om. Denna del är främst till för de som håller i lektionen. Därefter följer själva uppgiften och lösningarna. Lektionens upplägg förklaras mer ingående i introduktionen på nästa sida.

Lektionen är tänkt att ta 2 h, med 10 min rast inkluderat. Förslag på upplägg:

Del 1 & 2: 55–65 min

Rast: 10 min.

Del 3 & 4: 45–55 min

Senast uppdaterad: 29/9-21

MATHIVATION & HANDELSRÅDET



1 Introduktion

(10 min)

1.1 Information och förberedelse inför lektionen

Välkommen till en lektion i programmering och matematik, gjord av Mathivation och Handelsrådet! Innehållet riktar sig till elever åk 8–9 och tar ca 2h inklusive rast. I linje med Mathivations pedagogik är innehållet fristående från (och mer avancerat än) grundskolans kunskapskrav. Syftet med detta är att eleverna får öva på problemlösning, angripa okända koncept och uthållighet.

Lektionen består av praktiska tillämpningar av matematik och programmering för att modellera ekonomiska samband. Eleverna behöver ha en egen dator med tillgång till internet för att använda programmeringsverktyget Scratch. Lektionens sal behöver även vara utrustad med projektor som utbildaren kan använda via sin dator.

Eleverna bör var bekväma med att programmera i Scratch sedan tidigare, ha kännedom om funktioner, grafer och koordinatsystem. Utbildaren behöver vara med att bekväm att programmera, felsöka och redovisa i Scratch. Även en allmän förståelse av programmering behövs, exempelvis hur ett program körs, hur värden hanteras med hjälp av variabler och olika metoder för att reglera programflödet ("if-sats", "while-loop" och "for-loop"). Detta är dock något som man till stor utsträckning kan läsa sig till i lektionsmaterialet. Det finns även material och instruktioner på Scratchs egen hemsida men även andra internetresurser. Mathivations egna guide till Scratch hittar du [här](#).

Om eleverna inte har scratch-konton sedan tidigare, kan du antingen 1) förbereda ett "klassrum" i scratch med elevkonton åt dem (läs mer [här](#) eller i vår egen guide [här](#)) eller 2) låta eleverna använda gäst-konton. De behöver då inte logga in på ett konto, men de kan heller inte spara eller dela sina projekt.

Uppgiften är att hjälpa ett skoföretag med förarbetet inför en lansering av en ny skomodell. För att göra detta behöver eleverna använda sig av programmering. Skoföretaget vill veta:

- Vilken prissättning på skomodellen ger störst vinst?
- Hur många budbilar för utkörning av skor från företagets lager är lönsamt, och hur många lagerhyllor med skor körs då ut?
- Vad blir efterfrågan för skorna vid en viss prissättning?

De matematiska funktionerna för dessa samband kan verka avancerade för eleverna, men med programmering går det att enkelt analysera dem. I början av lektionen lär sig eleverna och övar på hur ett program är uppbyggt, hur variabler hanteras i Scratch och hur programmet effektiviseras med loopar och villkors-satser.

1.2 Väck elevernas intresse

Börja lektionen med att visa bildpresentationen och som presenterar Handelsrådet. Om man vill kan man använda bildpresentationen som stöd under hela lektionen. Presentation hittar du [här](#).

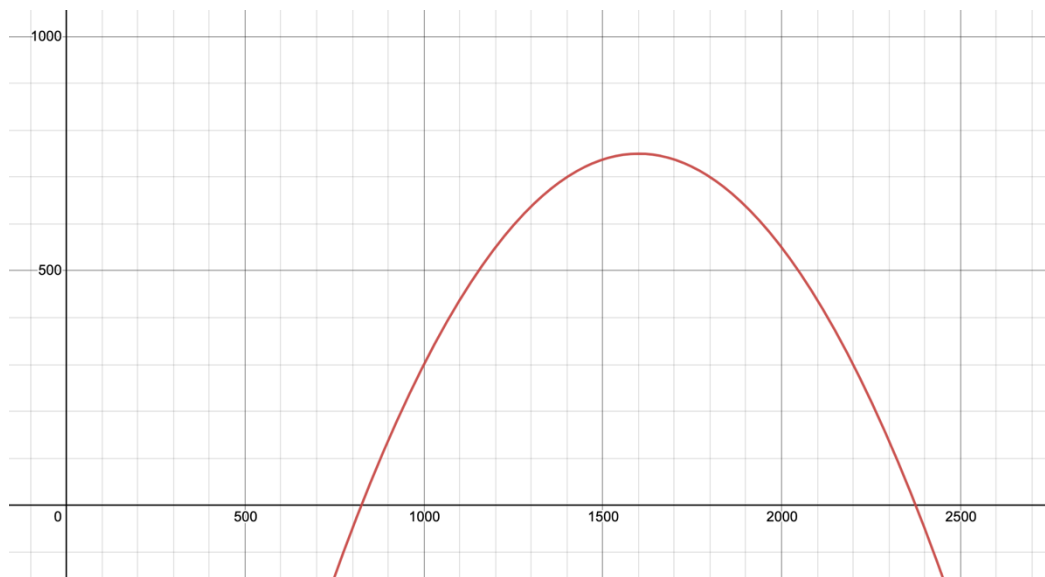
Berätta därefter att eleverna kommer att vara rådgivare till ett skoföretag som säljer sneakers. Företaget beställer skor från en tillverkare i Vietnam som skickas till ett lager, för att sedan säljas i webbshop och fraktas ut med budbilar. Företaget behöver nu hjälp med att 1) sätta pris för en ny modell och 2) ta reda på hur många budbilar deras lager behöver och 3) veta vilken efterfrågan det finns för skorna – för att inte beställa för många. För att undersöka ekonomiska problem som dessa ska vi använda både matematik och programmering. Mer exakt, för att hitta de bästa valen ska vi använda metoder för optimering.

2 Hitta optimalt pris för maximerad vinst

Totalt 55 min

2.1 Information till dig som håller lektionen

Eleverna ska skriva ett program i Scratch som analyserar en matematisk funktion, som beskriver hur vinsten beror på prissättningen av skorna. Programmet som eleverna programmerar ska beräkna funktionens värde, d.v.s. vinsten (variabel y) för olika skopris (variabel x). För att hitta det optimala priset som ger den maximala vinsten testar vi olika prisnivåer mellan 500 kr och 2 500 kr – vi testar priser i intervallet 500 – 2500 kr helt enkelt. Det pris som ger den högsta vinsten är det optimala priset. Genom att kolla på Figur 1 kan vi se att grafen är formad som ett upp-och-nedvänt U, där ett högre pris ger en större vinst fram till en viss punkt, kallad maxpunkt. Efter maxpunkten vänder det nedåt och ett ännu högre pris ger en lägre vinst. Vi kan avläsa att maxpunkten har priset (x) 1 600 kr vilket ger en vinst (y) på 750 kr. OBS! Avslöja **inte** maxpunktens x - och y -värde, det ska eleverna räkna ut själva.



Figur 1: Graf för funktion av vinst i kronor per sko (y) beroende på skomodellens pris i kronor (x).

För att identifiera funktionens max-värde (d.v.s. värdet som maxpunkten har) ska programmet spara det hittills högsta y -värdet. Detta jämförs sedan med varje nytt y -värde då programmet stegvis beräknar funktionen för alla olika x -värden. När ett nytt y -värde är högre än det sparade, ska programmet ersätta det sparade y -värdet med det aktuella nya y -värdet. En enkel förklaring är att programmet håller koll på "rekordet" och sparar koordinaterna (x , y) för den hittills högsta y -värdet. När programmet undersökt och jämfört alla y -värden (vinst) för x -värden inom intervallet 500 – 2500 kr (prissättning) kan den

besvara vilken max-punkt funktionen har. Punktens x-värde är alltså den optimala prissättningen på skorna.

2.2 Introduktion

(5 min)

Börja med att rita upp grafen ovan på tavlan med stora steg på y-axeln (markera endast 500, 1000, 1500 ...). Lämna medvetet steg längs x-axeln förutom i början och slutet (endast 0, 200, 400, 2800, 3000). Detta är för att eleverna **inte** ska kunna läsa av maxpunktens x-koordinater.

Förklara att grafen visar vinsten för skorna beroende på pris. Företaget behöver ett pris som täcker kostnaderna för produktion, personal, distribution och annat. Men enligt grafen avtar vinsten vid ett visst pris. Varför? - Jo, allt färre kunder är villiga att köpa skorna ju dyrare de är och därför minskar försäljningen.

Skriv därefter upp grafens funktion och fråga eleverna var det optimala priset finns i grafen. Funktionen för vinst ges av:

$$y = \frac{x}{2} - \frac{(x - 1400)^2}{800}$$

Fråga om eleverna har förslag på hur de kan ta reda på vilka x- och y-värden punkten har. För tydlighet, rita ett streckat streck lodrät ned från maxpunkten. Diskutera deras idéer, men berätta att vi nu ska ta hjälp av programmering och skriva ett program som beräknar svaret åt oss.

2.3 Att tänka som ett program

(15 min)

Förklara att ett program läser sin kod uppifrån och ned. Därför behöver vi dela upp vår lösning ("tänkandet") i olika steg. Eleverna ska nu arbeta med några korta uppgifter. Skriv upp exempel på funktioner på tavlan och be eleverna skriva ner vilka steg de gör för att beräkna ut funktionsvärdet (y). Uppgiften är alltså inte vad y är, utan att kunna förklara hur (vilka steg, i vilken ordning) lösningen för y kan se ut.

Förklara att vid varje nästa steg i "tänkandet" behöver man komma ihåg vilket värde man ska fortsätta räkna med. Vi gör detta naturligt, men ett program behöver göra det med hjälp av variabler. En förklarande bild är att se en variabel som ett utrymme (låda) i minnet, där vi kan lagra vilket värde vi vill för att sedan använda eller ändra just det värdet. För att få svaret av ett program behöver det alltså visa upp värdet av en variabel (innehållet i svars-lådan).

2.3.1 Övningar

Arbeta med uppgifterna A-F och be eleverna beskriva hur de använder minnes-lådorna.

A. $y = 4x - 20$

C. $y = x^2 + 20x$

E. $y = 9^x - \frac{20}{(x-8)}$

B. $y = 60^3 + 16 \cdot (4 - 2)$

D. $y = 4x^{10} + 3 - \frac{5}{x}$

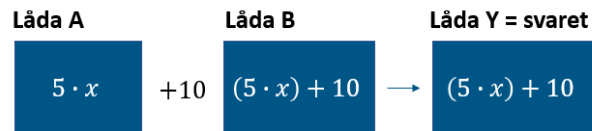
F. $y = x!$ (om $x = 5$, $y = 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1$)

Ge eleverna ca 10min innan de presenterar de sina lösningar (muntligt eller på skriftligt på tavlan). Se till att eleverna åtminstone får se en korrekt metod till uppgift A, C och E.

Om eleverna är fast kan du visa ett exempelsvar till funktionen $y = 5x + 10$. För att illustrera variabler kan du rita olika lådor på tavlan:

Exempel:

$$\triangleright y = 5x + 10$$



Här kan lösningen vara att beskriva stegen som:

1. Börja med att multiplicera x med 5 och spara produkten i låda A
2. Addera 10 till värdet som finns i låda A och spara i låda B
3. Plocka upp värdet i låda B och presentera det som svaret y (kan ritas som en låda Y)

Det kan verka onödigt att spara allt i Låda B, i stället för att bara *ändra* värdet i låda A med 10. Men man kanske behöver Låda B i ett senare skede och därför är det bra att öva på att skapa variabler. Om man vill kan utbildaren visa att båda sätten ger rätt svar (d.v.s. (1.) där vi lägger till ett värde och skapar en ny låda eller (2.) där vi ändrar värdet i en låda).

$$\text{Låda A} + \text{Låda B} = Y \quad (1.)$$

$$\text{Låda A} + 10 = Y \quad (2.)$$

2.3.2 Förslag på svar:

De minnes-utrymmen vi använder för variablerna kan kallas låda A, B, C men även "minne" eller "summa". I programmering är det upp till skaparen! Nedan ges förslag på svar som presenteras muntligt.

A.

1. 4 multiplicerat med x till ett minne
2. Subtrahera det med 20

B.

1. 4 minus 2 till låda A
2. 16 multiplicerat med låda A till låda B
3. Addera sedan produkten av $60 \cdot 60 \cdot 60$ till låda B
4. Presentera innehållet i låda B som svaret

C.

1. x multiplicerat med x sparas i ett minne
2. Innehållet i minnet adderas med 20 stycken x

D.

1. x multiplicerat med sig självt 10 gånger till låda A
2. 4 multiplicerat med låda A till låda B
3. Addera 3 till låda B
4. Subtrahera $\frac{5}{x}$ från låda B

E.

1. 9 multiplicerat med sig självt x gånger. Spara svaret i låda A
2. Ta $x-8$ och sedan 20 dividerat med det, spara i låda B
3. Subtrahera låda B från låda A

F.

1. Ta 1 multiplicerat med 2 och spara som minne
2. Ta sedan minnet multiplicerat med 3 och spara som nytt minne
3. Ta sedan nya minnet multiplicerat med 4 och spara som nytt minne
4. Fortsätt på samma sätt tills svaret multiplicerat med värdet på x

2.4 Programmera uppgift A - C (5 min)

Det är dags att börja programmera! Eleverna ska nu ta fram sina datorer och presentationen hålls i fortsättningen via projektor.

2.4.1 Förberedelse och introduktion till Scratch

Visa hur variabler skapas och hanteras i Scratch. Demonstrera hur funktionen A ovan kan skrivas i Scratch-kod, och hur programmet kan visa vad resultatet blir.

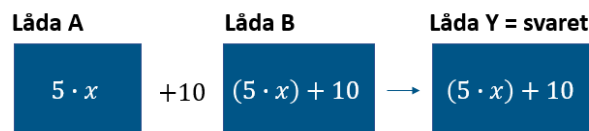
Arbeta tillsammans (be om hjälp och idéer via handuppräkning) med att skriva kod till funktionerna i tidigare uppgifterna B och C. Se lösningsexempel på kod nedan.

2.4.2 Skillnaden mellan kommandot ”sätt” och ”ändra” i Scratch

Vad är då skillnaden mellan kommandot ”sätt” och ”ändra” i Scratch? Jo, ”sätt” innebär att vi programmerar om det som tidigare stod där, d.v.s. det värde som vi säger att variabeln ska få får den oavsett vilket värde den hade innan – vi skriver helt enkelt över tidigare värde. Om vi tittar på exempeluppgiften vi såg förut:

Exempel:

➤ $y = 5x + 10$



Här skapar vi två olika lådor: *Låda A* + *Låda B* = *Y*. För att få fram svaret *Y* måste *Y* ”sättas” till *Låda B*.

I Scratch blir det:

1. ”sätt” *Låda A* till $5 \cdot x$,
2. ”sätt” *Låda B* till *Låda A* + 10,
3. ”sätt” *Låda Y* till *Låda B*.

Om vi kollar på kommandot ”ändra” så innebär detta att det befintliga värdet *ändras* med detta värde, d.v.s. vi kan exempelvis addera/subtrahera till/från det ursprungliga värdet. Vi behöver då inte skapa *Låda B*. Det här kan vara en enklare lösning, men det är bra lära sig båda ifall man i ett senare skede behöver använda sig av fler variabler.

I Scratch blir det:

1. ”sätt” *Låda A* till $5 \cdot x$,
2. ”ändra” *Låda A* med 10,
3. ”sätt” *Låda Y* till *Låda A*.

2.4.3 Lösningsförslag

Här kommer ett lösningförslag för varje uppgift. Men det finns även andra sätt att lösa uppgifterna på – det är det som är så roligt med programmering! Se alla lösningförslag i modulen som just förslag.

```

när a tangenten trycks ned
  sätt y till 4 * x
  ändra y med -20
  säg y i 2 sekunder
  
```

Uppgift A

```

när b tangenten trycks ned
  sätt var till 60 * 60 * 60
  ändra var med 16 * 4 - 2
  säg var i 2 sekunder
  
```

Uppgift B

```

när c tangenten trycks ned
  sätt y till x * x
  ändra y med 20 * x
  säg y i 2 sekunder
  
```

Uppgift C

2.5 Repetition av kod (5 min)

Förklara att man vill skriva kod som är så kort som möjligt, för att snabbt få ut svaret – vi vill ju testa 2000 olika beräkningar av x för att ta reda på skopriset! Visa att uppgifterna D eller E kan skrivas med en loop som repeterar samma kod. Demonstrera i Scratch samtidigt som eleverna härmar på sin dator när du skriver kod till båda funktionerna. Se lösningsexempel på kod nedan.

2.5.1 Lösningförslag

```

när d tangenten trycks ned
  repetera 9
    sätt var till x * x
    sätt var till 4 + var
    ändra var med 3 + 5 / x
  säg var i 2 sekunder
  
```

Uppgift D

```

när e tangenten trycks ned
  sätt y till 9
  repetera x - 1
    sätt y till y * 9
    sätt var till -20 / x - 8
    ändra y med var
  säg y i 2 sekunder
  
```

Uppgift E

```

när f tangenten trycks ned
  sätt var till x
  repetera x - 1
    ändra x med -1
    sätt var till var * x
  säg var i 2 sekunder
  
```

Uppgift F

2.6 Hitta det optimala priset för maximal vinst med hjälp av kod (20 min)

Nu när eleverna har fått testa på att skriva egen kod är det dags att gå tillbaka till själva uppgiften. Vi vill ju hitta det optimala priset för att få ut maximal vinst, med hjälp av programmering! Vi går därför tillbaka till funktionen för prissättning och vinst.

2.6.1 Tillgänglig information

Skriv upp på tavlan:

Funktionen:

$$y = \frac{x}{2} - \frac{(x - 1400)^2}{800}$$

och vi vill testa den för x mellan 500 – 2500.

Delsteg:

1. Skriv kod som räknar ut y av funktionen $y = \frac{x}{2} - \frac{(x-1400)^2}{800}$
2. Gör så att koden upprepas för x mellan 500 – 2500.
3. Lägg till kod så att programmet vet vilken punkt som hade störst y-värde!

2.6.2 Lösningen i delsteg

Låt eleverna arbeta med uppgiften ensamma eller i par. Eleverna skriver sin kod i egna, nya, Scratch-projekt. Gå runt och hjälp vid behov. Låt eleverna vara helt vilse och utforska problemet själva innan du ger någon ledtråd (5–10 min). Guida dem mot rätt väg genom att ge tips och instruktioner med jämna mellanrum. Observera att det är en del av uppgiften att eleverna själva ska lista ut när de ska använda kommandot ”ändra” och enbart ändra en variabel de redan har och när de bör använda ”sätt” och skapa en ny variabel.

Gå igenom lösningen till delstegen efter hand när alla elever fått fundera länge själva. När ni kommit fram till den slutliga lösningen till problemet, summera och spalta upp i korta punkter vilka delsteg som gjordes.

Ledtråd till delsteg 3: Spara rekordet av y-värdet och dess x-värde.

2.6.3 Villkor & skillnaden mellan kommandot ”repetera tills x” och ”om x då” i Scratch

Vad är då skillnaden mellan kommandot ”repetera tills x” och ”om x då” i Scratch? Jo, det har att göra med villkor. Ett villkor är en programmeringsterm som ställer ett påstående där man lägger in sitt påstående i luckan (x). Beroende på om det påståendet är uppfyllt är inte så kommer programmet att köras.



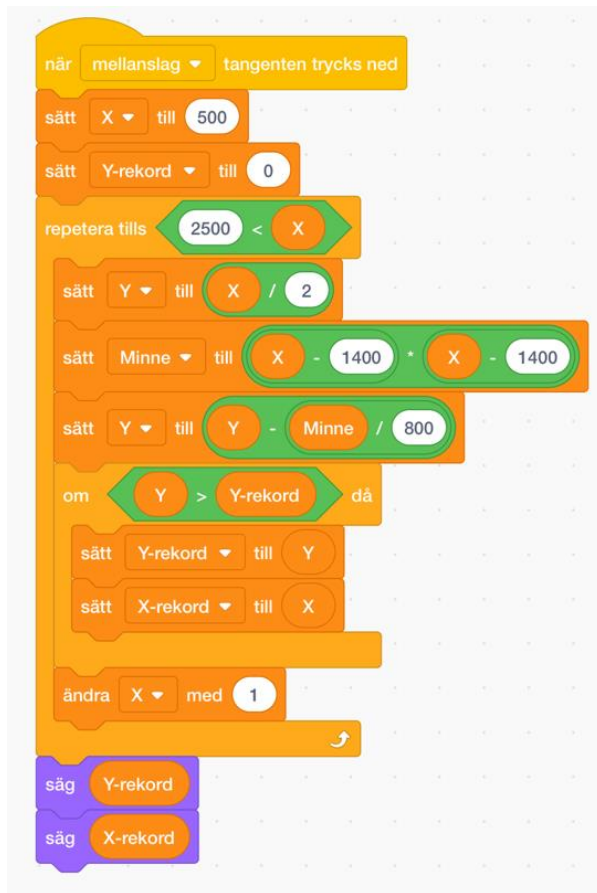
Så när vi använder ”repetera tills x” vill vi att programmet ska repetera koden inuti ”klämman” (d.v.s. funktionen) tills villkoret är uppfyllt. Detta innebär att programmet repeterar så länge som villkoret är falskt.



Om vi i stället kollar på kommandot ”om x då”, så är detta också ett villkor som ställer ett påstående där man lägger in sitt påstående i luckan (x). Här vill vi att koden inuti ”klämman” enbart ska köras om påståendet stämmer, d.v.s. om villkoret är sant. Om villkoret inte är sant (d.v.s. är falskt) hoppas koden över. Detta skiljer sig alltså från ”repetera tills”-kommandot som enbart körs så länge villkoret är falskt.

Kommandot "repetera" som användes i en tidigare uppgift är inget villkor, utan den kör bara koden så många gånger som man bestämt.

2.6.4 Lösningsförslag och svar



Svar: Maxpunkten finns då $x = 1600$ och $y = 750$.

Varför sätter vi Y-rekord till 0 i början av koden? (Se tredje raden i koden ovan.) Jo, det gör vi för att säkerställa att det inte ligger ett annat Y-rekord sparat. Vi sätter alltså Y-rekord till 0 för att nollställa alla parametrar till deras startvärde.

2.7 Redovisning (5 min)

Efter 20 minuter presenterar eleverna sina lösningar till uppgift 1. De kan antingen redovisa genom att dela sitt Scratch-projekt till den som håller i lektionen eller koppla upp sin dator i klassrummet (om enklare). Förklara vad lösningen, maxpunkten ($x = 1600$, $y = 750$) innebär för skoföretaget.

2.7.1 Slutsats

När priset är 1600 kr är skorna som mest lönsamma för företaget, vinsten blir då 750 kr.

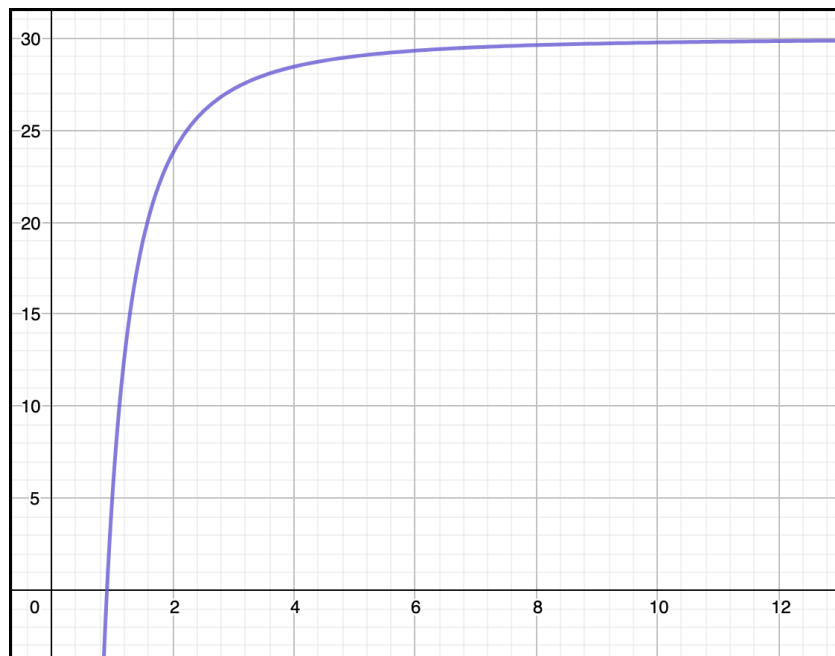
3 Hitta optimal kapacitet för budbilar

Totalt 25 min

3.1 Information till dig som håller lektionen

Okej, så nu vet vi det optimala priset för att få ut maximal vinst. En annan sak som är bra att veta är hur många budbilar som är det optimala antalet för att köra ut alla skor från lagret. Företaget vill även veta hur många lagerhyllor med skor som de kan köra ut, givet det optimala antalet budbilar. Vi ska därför börja med att kolla på en funktion som beskriver hur kapaciteten för att köra ut skor (y) beror på antalet

budbilar (x), för x mellan 1 – 15 (d.v.s. om vi har 1, 2... upp till 15 budbilar). Genom att kolla på Figur 3 kan vi se att grafen hastigt stiger uppåt för att sedan plana ut och att fler budbilar ger en större kapacitet fram till en viss punkt – där börjar kurvan avta för att sedan plana ut. Efter att kurvan börjat plana ut spelar det i princip ingen roll om vi köper in fler budbilar då vi ändå inte får ut en högre kapacitet. Ekonomer kallar detta fenomen för *lagen om avtagande avkastning*.



Figur 2: Graf av funktion för antal lagerhyllor med skor utkörda per vecka (y) beroende på antal budbilar (x)

I den här uppgiften ska alltså eleverna skriva ett program som undersöker en funktion som beskriver hur kapaciteten att köra ut skor (y) (där y = antal lagerhyllor med skor som hinner köras ut per vecka) beror på antalet budbilar (x), för x mellan 1 – 15. Algoritmen ska spara de senaste y -värdena och därefter jämföra hur stor skillnaden är – jämfört med föregående y -värde. Om ökningen mellan två y -värden är mindre än 0,4 (vilket är vårt förbestämda tröskelvärde, $K = 0,4$) har kurvan börjat plana ut och vi vill inte längre köpa in fler budbilar.

3.2 Introduktion (5 min)

Rita först upp grafen ovan på tavlan. Skriv ut värden på y -axeln men endast 0, 1, 2 och hoppa sedan direkt till 15 på x -axeln. Förklara sedan att den visar hur antalet varor som kan köras ut från företagets lager ökar ju fler budbilar vi använder. Det syns att skillnaden i ökningen avtar och till slut hjälper det inte att fler budbilar används. Fråga om eleverna kan lista ut varför? – Jo, ökningen begränsas till slut av utrymme vid lastbryggan, personalen hinner inte packa fler budbilar, trängsel som tar mer tid etc. Det blir en flaskhals som gör att det inte hjälper med fler budbilar.

3.3 Hitta det optimala antalet budbilar för maximal kapacitet (15 min)

3.3.1 Tillgänglig information

Företaget anser att om en ny budbil ökar antalet utkörningar med mindre än 0,4 hyllor från lagret (motsvarande 0,4 skillnad mellan ett funktionsvärde och nästa) är den inte lönsam.

Funktionen för y lagerhyllor med skor utkörda per vecka av x budbilar är:

$$y = 30 - \frac{36}{x^2}$$

och vi vill testa den för x mellan 1 – 15.

Skriv upp delstegen för uppgiften:

1. Skriv kod som beräknar y till funktionen $y = 30 - \frac{36}{x^2}$
2. Gör så att koden upprepas för alla x mellan 1 – 15
3. Hur ska programmet förstå när nästa lastbil blir onödig? Skriv kod som upptäcker vid vilket x skillnaden av y är mindre än 0,4.
4. Använd det optimala antalet budbilar för att få fram antalet lagerhyllor som körs ut.

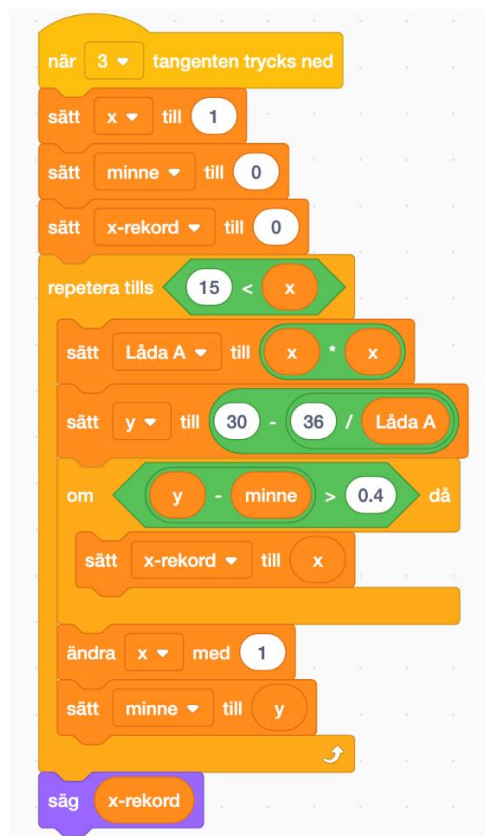
3.3.2 Fråga

Företaget undrar nu hur många budbilar som är optimalt? De vill även veta hur många lagerhyllor med skor som de kan köra ut, givet det optimala antalet budbilar.

3.3.3 Lösningen i delsteg

Låt eleverna arbeta med uppgiften ensamma eller i par. Gå runt och hjälp vid behov. Låt eleverna vara helt vilse och utforska problemet själva innan du ger någon ledtråd (5–10 min). Guida dem mot rätt väg genom att ge mer information med jämna mellanrum. Gå igenom lösningen till delstegen efter hand när alla elever fått fundera länge själva. När ni kommit fram till den slutliga lösningen till problemet, summera och spalta upp i korta punkter vilka delsteg som gjordes.

3.3.4 Lösningsförslag och svar



x 16
minne 29.84
x-rekord 6
låda a 225
y 29.84



Svar: Det optimala antalet budbilar är sex då $x = 6$, vilket ger $y = 29$.

Om vi programmerat rätt i Scratch kommer vi få fram att vårt x -rekord är 6, d.v.s. att $x = 6$, vilket innebär att sex budbilar ger optimal kapacitet. För att få fram hur många lagerhyllor med skor som vi då kör ut sätter vi in vårt x -rekord i funktionen:

$$y = 30 - \frac{36}{x^2} \Rightarrow y = 30 - \frac{36}{6^2} \Rightarrow 30 - \frac{36}{36} \Rightarrow 30 - 1 = 29$$

Vi får då fram att $y = 29$. Visst hade företaget kunnat köra ut ännu fler lagerhyllor med skor om de hade haft fler budbilar, men det hade inte varit lönsamt för företaget. Hur vet vi då att det inte hade varit lönsamt? Jo, för ytterligare en budbil hade ökat antalet utkörningar med mindre än 0,4 lagerhyllor.

OBS! Scratch kommer även visa ett y -värde som är 29,84. Detta är *inte* det y -värdet som vi får fram om vi använder oss av vårt x -rekord-värde, vilket är 29. Utan det här y -värdet (29,84) är det sista y -värdet i algoritmen, d.v.s. när $x = 15$ (detta då vi brett programmet att testa alla x från 1-15). Om vi sätter in $x = 15$ i funktionen får vi:

$$y = 30 - \frac{36}{x^2} \Rightarrow y = 30 - \frac{36}{15^2} \Rightarrow 30 - \frac{36}{225} \Rightarrow 30 - 0,16 = 29,84$$

Men varför testade vi inte för alla x mellan 0 – 15, i stället för 1 – 15? Varför kunde vi inte testa från 0? (Se andra raden i koden ovan där x är satt till 1). Jo, för det första så vill ju företaget ha budbilar, så noll budbilar är inget alternativ. För det andra, så om vi kollar på funktionens uppbyggnad så har vi x i nämnaren och vi kan inte dividera med 0.

3.3.4.1 Om det behövs ytterligare förklaring:

För tydlighets skull kan vi även testa med fyra budbilar, då får vi:

$$y = 30 - \frac{36}{x^2} \Rightarrow y = 30 - \frac{36}{4^2} \Rightarrow 30 - \frac{36}{16} \Rightarrow 30 - 2,25 = 27,75$$

Och med fem budbilar får vi:

$$y = 30 - \frac{36}{x^2} \Rightarrow y = 30 - \frac{36}{5^2} \Rightarrow 30 - \frac{36}{25} \Rightarrow 30 - 1,44 = 28,56$$

Vi ser då att förändringen mellan y -värdet 27,75 (givet $x = 4$) och y -värdet 28,56 (givet $x = 5$) är 0,81, vilket är över tröskelvärdet 0,4. Förändringen mellan y -värdet 28,56 (givet $x = 5$) och y -värdet 29 (givet $x = 6$) är 0,44, vilket också är över tröskelvärdet 0,4.

Om vi i stället testar med sju budbilar får vi:

$$y = 30 - \frac{36}{x^2} \Rightarrow y = 30 - \frac{36}{7^2} \Rightarrow 30 - \frac{36}{49} \Rightarrow 30 - 0,73 = 29,27$$

Vi ser då att förändringen mellan y -värdet 29 (givet $x = 6$) och y -värdet 29,27 (givet $x = 7$) är 0,27, vilket är mindre än tröskelvärdet 0,4. Vi kan därför återigen konstatera att sex budbilar är det optimala antalet för företaget, för fler än sex budbilar ökar inte antalet utkörningar tillräckligt mycket (förändringen i y är mindre än 0,4).

3.4 Redovisning

(5 min)

Eleverna presenterar sina lösningar efter 20 minuter. De kan antingen redovisa genom att dela sitt Scratch-projekt till den som håller i lektionen eller koppla upp sin dator i klassrummet (om enklare). Förklara vad lösningen ($x = 6$, $y = 29$) innebär för skoföretaget.

3.4.1 Slutsats

I den här uppgiften var vi först ute efter x -rekord-värdet som vi fick fram genom att kolla på hur förändringen i y -värdet (d.v.s. skillnaden mellan två y -värden, även kallat delta- y), förhöll sig till tröskelvärdet 0,4, där företaget bestämt att om skillnaden var mindre än 0,4 så ville de inte ta in fler

budbilar. Vi fick fram att sex budbilar ger optimal kapacitet ($x = 6$) för lagret. Har företaget fler än sex budbilar ökar de inte utkörningen tillräckligt mycket för att vara lönsamma (varje ny budbil tillför en kostnad). Med sex budbilar kan företaget leverera 29 lagerhyllor av skomodellen per vecka.

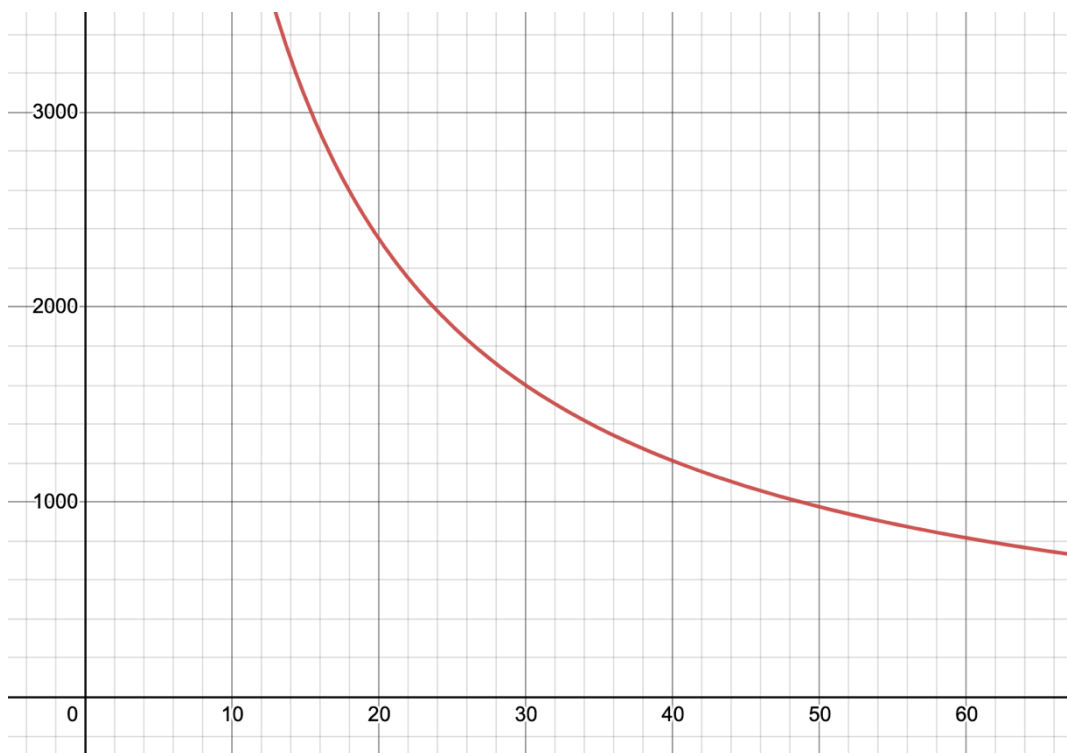
4 Hitta efterfrågan vid optimalt pris

Totalt 20 min

4.1 Information till dig som håller lektionen

Okej, så nu vet vi även hur många budbilar som är det optimala antalet för företaget. En annan sak som är bra att veta är hur efterfrågan på skorna förändras när priset förändras. I tidigare uppgift kom vi fram till det optimala priset utifrån maximal vinst, men vi vet fortfarande inte hur många skor (lagerhyllor) som vi säljer.

Vi ska därför kolla på en funktion som beskriver hur skornas pris (y) påverkar den veckovisa efterfrågan på lagerhyllor med skor (x). Figur 4 visar efterfrågekurvan över hur många lagerhyllor med skor som konsumenterna vill köpa vid olika priser. Vid ett lägre pris vill konsumenterna köpa många lagerhyllor med skor och vid ett högre pris vill de inte köpa lika många lagerhyllor med skor. Då konsumenterna vill köpa färre lagerhyllor med skor när priset ökar lutar efterfrågekurvan nedåt.



Figur 3: Graf av funktion för pris på skomodell (y) beroende på efterfrågan på lagerhyllor med skor per vecka (x)

Om det är mindre än 20 min kvar av lektionen kan den här uppgiften förkortas genom att eleverna löser uppgiften för hand eller med hjälp av en miniräknare i stället för att skriva ett program i Scratch som visar svaret.

4.2 Introduktion

(5 min)

Rita först upp grafen ovan på tavlan. Skriv ut värden på y-axeln men endast 0, 1, 2 och hoppa sedan direkt till 39, 40 på x-axeln. Förklara sedan att den visar hur efterfrågan av lagerhyllor med skor per vecka beroende på hur mycket skomodellen kostar. Ju högre pris, desto lägre efterfrågan från kunder. Vi vill nu veta hur stor efterfrågan är när skorna kostar 1 600 kr, vilket var det optimala priset som vi kom fram till i del 2.

4.3 Hitta efterfrågan på lagerhyllor med skor vid optimalt pris

(10 min)

4.3.1 Tillgänglig information

Optimalt pris (från del 2): 1 600 kr

Funktionen för priset y på skomodellen som ger efterfrågan x av skor som efterfrågas per vecka:

$$y = \frac{50\,000}{(x + 1,25)}$$

4.3.2 Fråga

Företaget undrar nu hur många lagerhyllor med skor som efterfrågas vid priset 1 600 kr.

Låt eleverna arbeta med uppgiften ensamma eller i par i 10 min. Gå runt och hjälp vid behov.

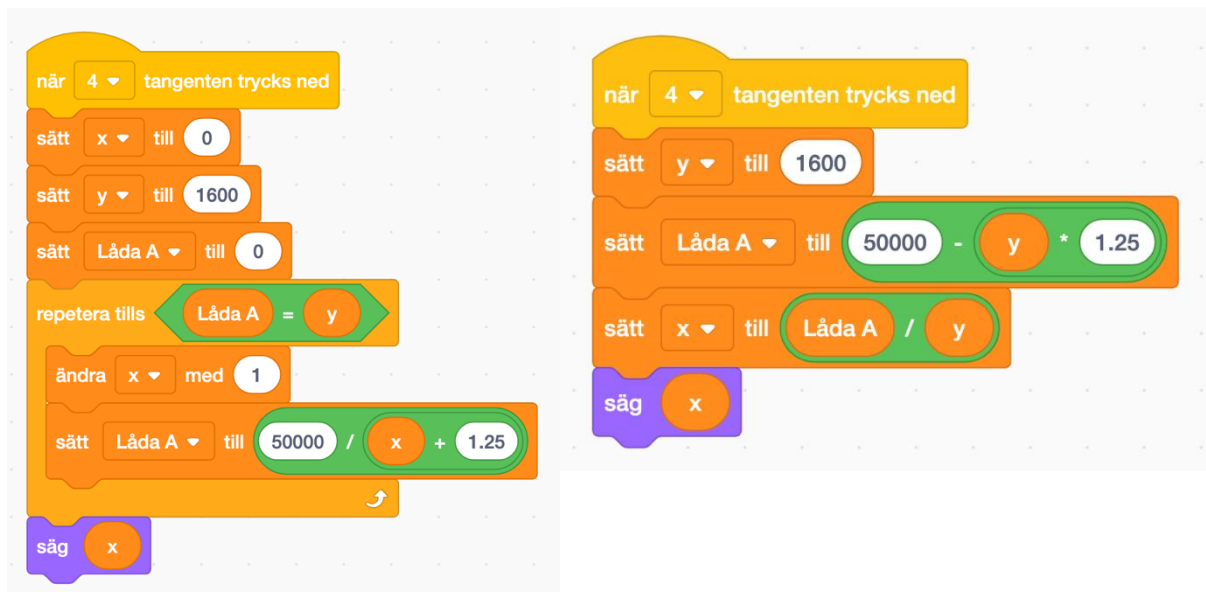
4.3.3 Lösningsförslag och svar

4.3.3.1 För hand eller miniräknare:

$$\begin{aligned} y = \frac{50\,000}{(x + 1,25)} \Rightarrow 1600 &= \frac{50\,000}{(x + 1,25)} \Rightarrow 1600 \cdot (x + 1,25) = 50\,000 \\ \Rightarrow 1600x &= 50\,000 - 1600 \cdot 1,25 = 48\,000 \Rightarrow x = \frac{48\,000}{1600} = 30 \end{aligned}$$

4.3.3.2 Genom att skriva ett program i Scratch:

Koden till vänster använder samma metod som tidigare uppgifter, d.v.s. den undersöker värden på x från 0 och uppåt. När funktionsvärdet (Låda A) är lika med vårt pris 1 600 kr, har vi hittat ekvationens svar på x . Koden till höger agerar däremot mer som en miniräknare, och förutsätter att eleven har skrivit om ekvationen så att x är fritt (likt lösningen för hand).



Svar: När priset är 1 600 kr efterfrågas 30st lagerhyllor med skor per vecka.

4.4 Redovisning

Om tid finns kan eleverna presentera sina lösningar, annars räcker det med att muntligt säga vilket värde de kom fram till ($x = 30$). Förklara att det innebär att skomodellens efterfrågan är 30 lagerhyllor per vecka. Detta innebär dock inte att vi säljer 30 lagerhyllor per vecka, utan vi vet ju sedan tidigare att vi endast har kapacitet för att leverera 29 lagerhyllor. Alltså är efterfrågan högre än vad vi kan leverera vid priset 1600 kr. Men då det är detta pris som ger maximal vinst så låter vi priset vara kvar på 1 600 kr.

Om de har löst uppgiften för hand eller med miniräknare är det kul om de kan komma fram till tavlan och visa sin lösning. Om de istället har löst uppgiften i Scratch kan de antingen redovisa genom att dela sitt Scratch-projekt till den som håller i lektionen eller koppla upp sin dator i klassrummet (om enklare).

4.5 Animering i Scratch

(5 min)

För att knyta ihop lektionen avslutas den med en sammanfattande uppgift där eleverna får göra en animeringsuppgift i Scratch. Börja med att sammanfatta vad eleverna har kommit fram till i de tidigare uppgifterna och skriv upp slutsatserna som ni vill presentera för skoföretaget:

1. Det optimala priset för skorna är 1 600 kr
2. 6 budbilar är optimalt, vilket ger att 29 lagerhyllor kan köras ut per vecka.
3. Efterfrågan för skor med priset 1600 kr är 30 lagerhyllor per vecka.

Visa på din skärm ett förberett Scratch-projekt som animerar företagets lager för skorna:

- <https://scratch.mit.edu/projects/449324576/>

För enkelhets skull illustreras lagerhyllor som kartonger. Eleverna behöver nu endast skriva in rätt värden för variablerna "Efterfrågan per vecka" (30) och "Utkörning per vecka" (29) och därefter se om illustrationen fungerar som den ska.

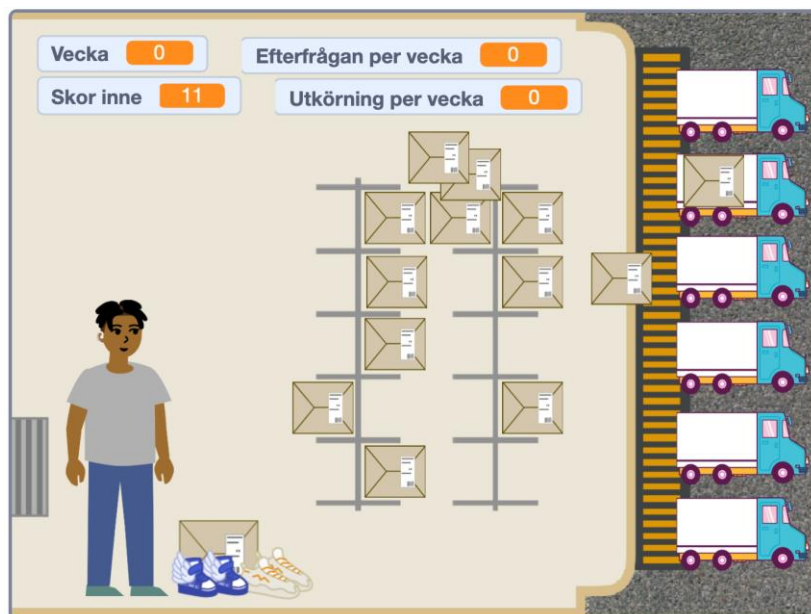


Bild på animationen i Scratch

4.5.1 Extrauppgift: remixa projektet och se hur vinsten ökar

Om det finns tid kvar kan eleverna lägga till egen kod i projektet som beräknar och visar hur vinsten ökar allt eftersom veckorna går. För att göra det behöver de först välja "remix" i Scratch (då sparas en kopia av deras version).

4.5.1.1 Tillgänglig information

- Funktion för vinst per sko finns från del 1: $y = \frac{x}{2} - \frac{(x-1400)^2}{800}$
- En hylla i lagret rymmer 12 par skor.
- Med 6 budbilar kan 29 lagerhyllor med skor köras ut per vecka.