Homework 11

Exercise 1

The present value of the two period income is: 5000 + 5500/1.10

```
5000 + 5500/(1.1)
```

```
10 000.
```

We choose the bundle (c_1, c_2) to maximize U such that $c_1 + c_2 = 10000$.

We set up the Lagrangian:

$\mathcal{L}(c_1, c_2, \lambda) = \ln(c_1) + \ln(c_2) + \lambda\, c_1 + c_2 - 10000$

$\frac{d\mathcal{L}}{d\lambda} = c_1 + c_2 - 10000 = 0$

$\frac{d\mathcal{L}}{dc_1} = \frac{1}{c_1} + \lambda\, c_1 = 0$

$\frac{d\mathcal{L}}{dc_2} = \frac{1}{c_2} + \lambda\, c_2 = 0$

```
Solve[c1 + c2 - 10 000 == 0 && 1/c1 + λ*c1 == 0 && 1/c2 + λ c2 == 0, {c1, c2, λ}]
```

$\{\{c1 \to 5000,\ c2 \to 5000,\ \lambda \to -\frac{1}{25\,000\,000}\}\}$

Or solving manually:

(first solve for λ)

$\frac{1}{c_1} + \lambda\, c_1 = 0$

$\lambda\, c_1 = \frac{-1}{c_1}$

$\lambda = \frac{-1}{c_1^2}$

(Substitute into second expression)

$\frac{1}{c_2} + \lambda\, c_2 = 0$

$\frac{1}{c_2} - \frac{c_2}{c_1^2} = 0$

$\frac{1}{c_2} = \frac{c_2}{c_1^2}$

$c_1^2 = c_2^2$

$\pm c_2 = \pm\, c_1$

(Substitute into constraint)

$c_1 + c_2 = 10000$

$2\, c_1 = 10000$ or $-2\, c_1 = 10000$

$(c_1, c_2) = (5000, 5000)$ or $(c_1, c_2) = (-5000, -5000)$

Second solution makes no economioc sense.

$\therefore (c_1, c_2) = (5000, 5000)$

Exercise 2

Definition: (y_n) is a *subsequence* of (x_n) $\exists_{\text{strictly increasing } f: \mathbb{N} \longmapsto \mathbb{N}} \forall_{n \in \mathbb{N}}\ y_n = x_{f(n)}$. That is, all members of (y_n) is an element of (x_n) obtained by removing some elements of (x_n) without changing order.

A sequence (x_t) in S converges to x ∈ S iff every subsequence of (x_t) converges to x.

Proof

Backward
$\forall_{\epsilon > 0}\ \exists_{n \in \mathbb{N}}\ d(x_n, \text{x}) < \epsilon$. (that is (x_t) converges to x)
$\because (y_t)$ is a subsequence of (x_t) any element y_n of (y_t) is the same as the element $x_{f(n)}$ of (x_t) (where f is strictly increasing)
$\therefore \forall_{\epsilon > 0}\ \exists_{n \in \mathbb{N}}\ d(y_n, \text{x}) < \epsilon$ (i.e. (y_t) converges to x)

Forward
$\forall_{\epsilon > 0}\ \exists_{n \in \mathbb{N}}\ d(y_n, \text{x}) < \epsilon$. (that is (y_t) converges to x)
$\because$ f is a strictly increasing function, there must exist an inverse function g = $f^{-1}\ \mathbb{N} \longmapsto \mathbb{N}$.
$\therefore$ any element x_n in the sequence (x_t) is the same as the element $y_{g(n)}$ in the subsequence (y_t).
$\therefore \forall_{\epsilon > 0}\ \exists_{n \in \mathbb{N}}\ d(x_n, \text{x}) < \epsilon$ (i.e. (y_t) converges to x)

Exercise 3

Prove $\lambda(\mathbf{x} + \mathbf{y}) = \lambda\mathbf{x} + \lambda\mathbf{y}$

Consider the vector $(\mathbf{x} + \mathbf{y})$. The sum of the j^{th} element is $(x_j + y_j)$. Multiplying the vector $(\mathbf{x} + \mathbf{y})$ by λ, we obtain the vector $\lambda(\mathbf{x} + \mathbf{y})$, whose j^{th} element is $\lambda(x_j + y_j) = \lambda \text{x}_j + \lambda \text{y}_j$ (by distributive property of multiplication over addition on the Reals).

Now consider the sum of the vector $\lambda\mathbf{x}$ and $\lambda\mathbf{y}$: $(\lambda\mathbf{x} + \lambda\mathbf{y})$. The j^{th} element of $\lambda\mathbf{x}$ is λx_j and of $\lambda\mathbf{y}$ is λy_j. Therefore the sum of the j^{th} element is $\lambda \text{x}_j + \lambda \text{y}_j$. Therefore the j^{th} row of $\lambda(\mathbf{x} + \mathbf{y})$ is the same as the j^{th} row of $\lambda\mathbf{x} + \lambda\mathbf{y}$. Since j is an arbitrary row, this is true for the entire vector sum.

Prove $(\lambda_1 + \lambda_2)\mathbf{x} = \lambda_1\mathbf{x} + \lambda_2\mathbf{x}$.

Consider the vector $\lambda_1\mathbf{x}$. Its j^{th} element is given by $\lambda_1\, x_j$. Similarly for the vector $\lambda_2\mathbf{x}$ the j^{th} element is given by $\lambda_1\, x_j$. Thus the j^{th} row of the sum of the vectors $\lambda_1\mathbf{x} + \lambda_2\mathbf{x}$ is given by: $\lambda_1\, x_j + \lambda_2\, x_j = \lambda_1 + \lambda_2\, x_j = (\lambda_1 + \lambda_2)\, x_j$ (by the distributive property of multiplication over addition on reals). Since x_j is arbitrary, this holds for the entire vector, that is $\lambda_1\mathbf{x} + \lambda_2\mathbf{x} = (\lambda_1 + \lambda_2)\ \mathbf{x}$.

Exercise 4

Base case
Let n = 1.
S(1) = 1 = 1^2.

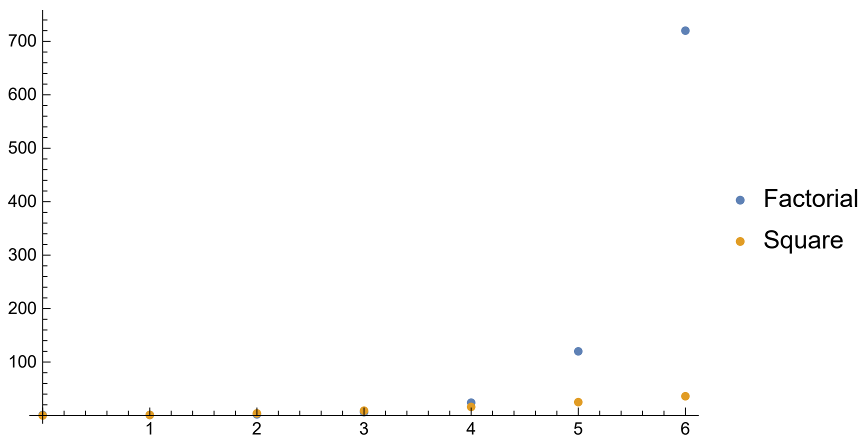
Recursive case
If S(n-1) = $(n-1)^2$
Then S(n) = $(n-1)^2$ + (2n - 1)
(n-1) (n-1) + 2n + 1 = n^2 - 2n + 1 + 2n - 1 = n^2

Computational Exercise 1

```
fact[0] := 1
fact[x_] := x * fact[x - 1]
```

```
fact[0]
fact[1]
fact[2]
fact[3]
fact[4]
facts = {#, fact[#]} & /@ Range[0, 6]
squares = { #, #^2} & /@ Range[0, 6]
ListPlot[{facts, squares}, PlotRange → All, PlotLegends → {"Factorial", "Square"}]
```

```
1

1

2

6

24

{{0, 1}, {1, 1}, {2, 2}, {3, 6}, {4, 24}, {5, 120}, {6, 720}}

{{0, 0}, {1, 1}, {2, 4}, {3, 9}, {4, 16}, {5, 25}, {6, 36}}
```



The factorial function grows faster

Computational Exercise 2

```
soln = Solve[1053.9 * Exp[rIndia * 10] == 1224.6 &&
    1269.1 * Exp[rChina * 10] == 1341.3, {rIndia, rChina}]
```

Solve::ifun :

Inverse functions are being used by Solve, so some solutions may not be found; use Reduce for complete solution information. >>

```
{{rIndia → 0.0150117, rChina → 0.00553313}}
```

```
Solve[(1053.9*Exp[rIndia*t] == 1269.1*Exp[rChina*t] /. soln), t]
```

Solve::ifun :
Inverse functions are being used by Solve, so some solutions may not be found; use Reduce for complete solution information. >>

```
{{t → 19.6033}}
```

i.e. by 2020 India’s population will overtake China.

```
soln = Solve[2.53*^9 * Exp[r*(2015 - 1950)] == 7.3*^9, r]
proj = 2.53*^9 Exp[r*(2050 - 1950)] /. soln
```

Solve::ifun :
Inverse functions are being used by Solve, so some solutions may not be found; use Reduce for complete solution information. >>

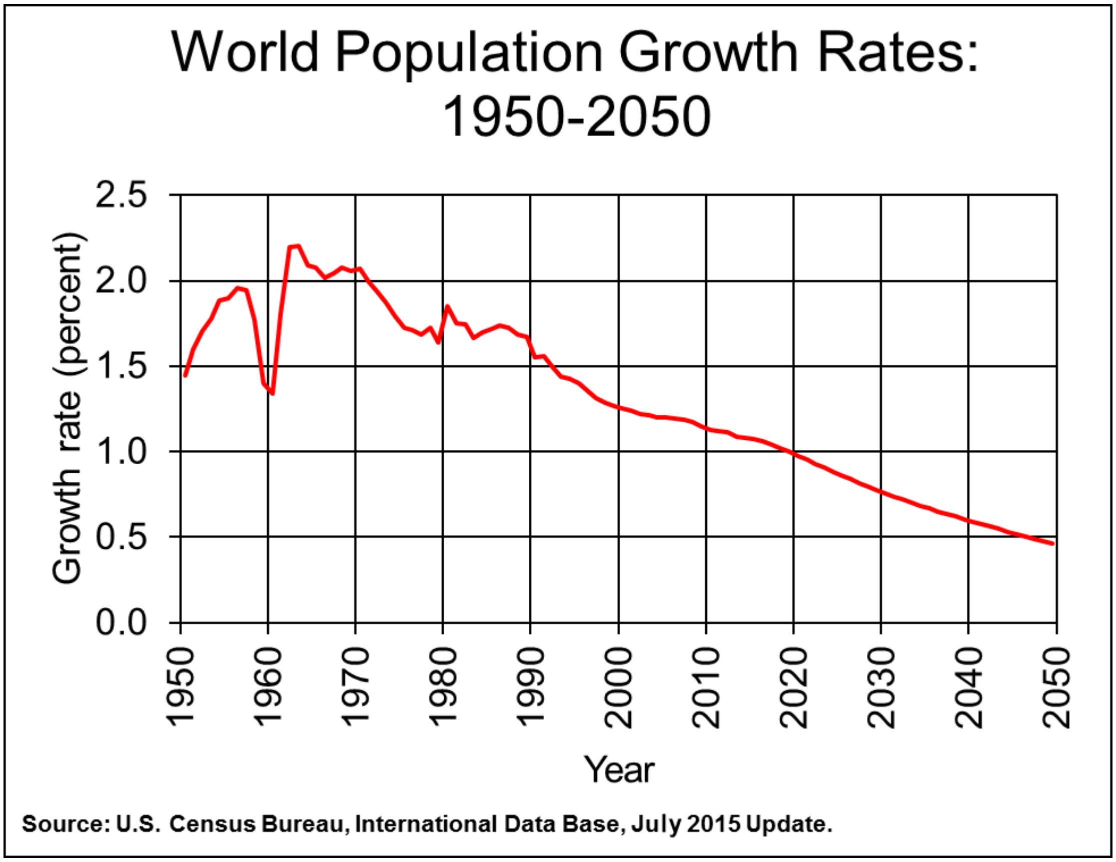
```
{{r → 0.0163024}}
```

$\{1.29159 \times 10^{10}\}$

i.e. 12.9 billion.

Carrying Capacity: the number of people, other living organisms, or crops that a region can support without environmental degradation. (source: dictionary.com)

Such a projection is not realistic. For example, while in the period before the 1980s the growth rate was higher than the average of 1.63%, growth rates have been steadily declining since then. Thus, the average growth rate over the next 50 years are going to be much lower than the average growth rate over the last 50 years. Here is the Census bureau’s projection on this:



Computational Exercise 4

Assume at t = 0 we start at q_0.
At t = 1, we are at:
$q_1 = 0.86\, q_0$.
At t = 2, we are at:
$q_2 = 0.86\, q_1 = 0.86^2\, q_0$
and so on
therefore we have at t=n
$q_n = 0.86^n\, q_0$

```
Solve[q0 * (0.86^n) == (1/2) q0, n]
```

Solve::ifun :
Inverse functions are being used by Solve, so some solutions may not be found; use Reduce for complete solution information. >>

```
{{n → 4.59577}}
```

Therefore, after about 4.6, we will be at half the value from the deviation.

Computational Exercise 5

the interest rate at which the net present value of all the cash flows from a project or investment equal zero.

```
project1 = Cashflow[{-100 000., 60 000, 47 000}]
project2 = Cashflow[{-100 000., 50 000, 40 000, 20 000}]
irr1 = Solve[{TimeValue[project1, i1, 0] == 0, i1 ≥ 0}, {i1}]
irr2 = Solve[{TimeValue[project2, i2, 0] == 0, i2 ≥ 0}, {i2}]
```

```
Cashflow[{-100 000., 60 000, 47 000}]
```

```
Cashflow[{-100 000., 50 000, 40 000, 20 000}]
```

Solve::ratnz : Solve was unable to solve the system with inexact coefficients.
The answer was obtained by solving a corresponding exact system and numericizing the result. ≫

```
{{i1 → 0.0483315}}
```

Solve::ratnz : Solve was unable to solve the system with inexact coefficients.
The answer was obtained by solving a corresponding exact system and numericizing the result. ≫

```
{{i2 → 0.0572597}}
```

Pick project that pays the stream {50000,40000, 20000} since this has a higher IRR (5.7% vs. for other project 4.8%).

```
project1a = Cashflow[{-90 000., 60 000, 47 000}]
project2a = Cashflow[{-90 000., 50 000, 40 000, 20 000}]
irr1 = Solve[{TimeValue[project1a, i1, 0] == 0, i1 ≥ 0}, {i1}]
irr2 = Solve[{TimeValue[project2a, i2, 0] == 0, i2 ≥ 0}, {i2}]
```

```
Cashflow[{-90 000., 60 000, 47 000}]
```

```
Cashflow[{-90 000., 50 000, 40 000, 20 000}]
```

Solve::ratnz : Solve was unable to solve the system with inexact coefficients.
The answer was obtained by solving a corresponding exact system and numericizing the result. ≫

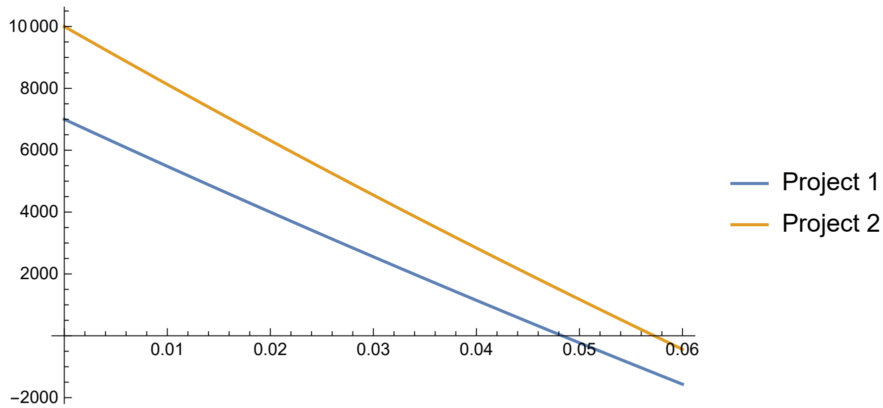
```
{{i1 → 0.129156}}
```

Solve::ratnz : Solve was unable to solve the system with inexact coefficients.
The answer was obtained by solving a corresponding exact system and numericizing the result. ≫

```
{{i2 → 0.125722}}
```

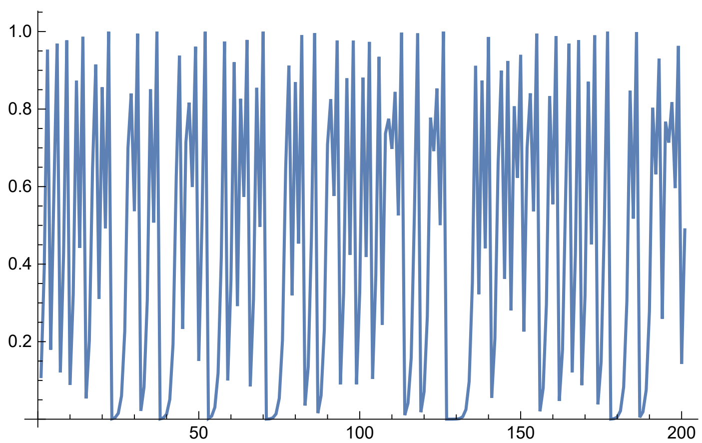
With a fall in the inital investment to 90000, the project paying {60000, 47000} now has a higher IRR (12.9% vs. other project which makes 12.6%).

```
Plot[{TimeValue[project1, r, 0], TimeValue[project2, r, 0]},
 {r, 0, 0.06}, PlotLegends → {"Project 1", "Project 2"}]
```



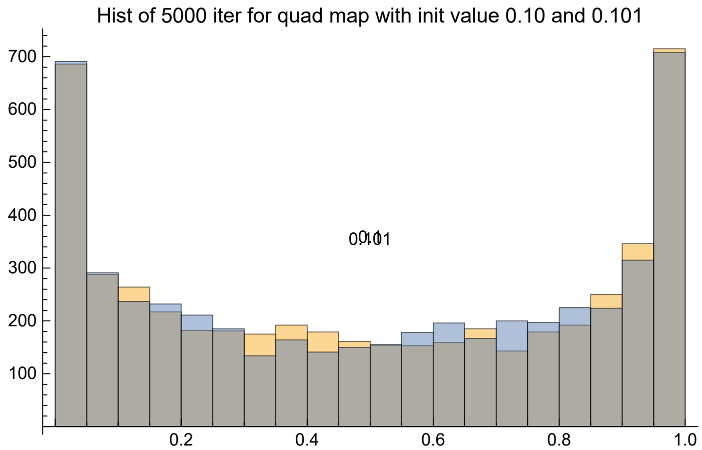
Computational Exercise 6

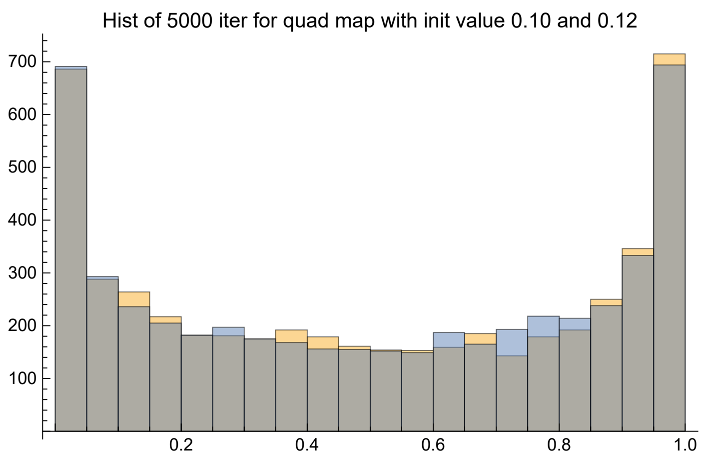
```
ListLinePlot[NestList[4 # (1 - #) &, 0.11, 200]]
```

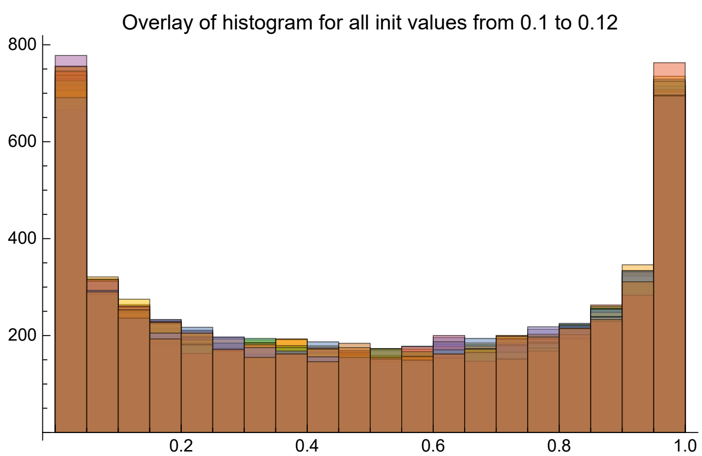


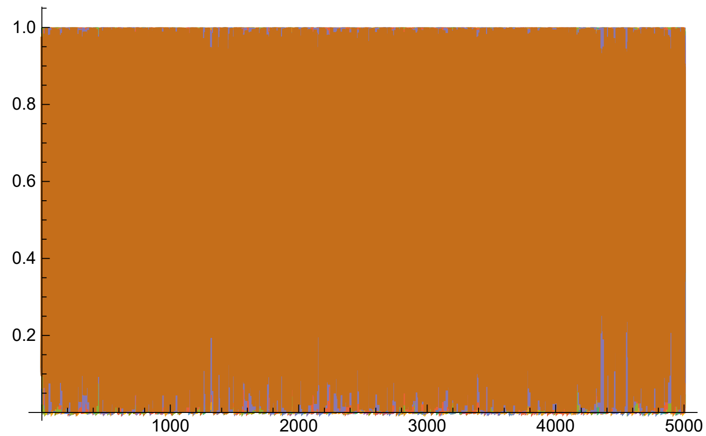
Computational Exercise 7

```
res = NestList[4 # (1 - #) &, #, 5000] & /@ Range[0.10, 0.12, .001];
Histogram[{res[[1]], res[[2]]},
 ChartLabels → Placed[Range[0.1, 0.12, 0.001], Center], PlotRange → All,
 PlotLabel → "Hist of 5000 iter for quad map with init value 0.10 and 0.101" ]
Histogram[{res[[1]], res[[20]]},
 PlotLabel → "Hist of 5000 iter for quad map with init value 0.10 and 0.12"]
Histogram[res, PlotLabel →
  "Overlay of histogram for all init values from 0.1 to 0.12"]
ListLinePlot[
 res]
```


Hist of 5000 iter for quad map with init value 0.10 and 0.101
700
600
500
400
300
200
100
0.1
0.101
0.2
0.4
0.6
0.8
1.0


Hist of 5000 iter for quad map with init value 0.10 and 0.12
700
600
500
400
300
200
100
0.2
0.4
0.6
0.8
1.0


Overlay of histogram for all init values from 0.1 to 0.12
800
600
400
200
0.2
0.4
0.6
0.8
1.0



Comments: We get every number between 0 and 1 but a bunching up near 0 and 1.

```
myEuclideanDistance[x_, y_] := Sqrt[Total[MapThread[(#2 - #1)^2 &, {y, x}]]]

myEuclideanDistance[{1}, {5}] (* test my function *)
EuclideanDistance[{1}, {5}] (* test built in function *)

myEuclideanDistance[{1, 2}, {5, 6}] (* test my function *)
EuclideanDistance[{1, 2}, {5, 6}]

myEuclideanDistance[{1, 2, 3}, {5, 6, 7}] (* test my function *)
EuclideanDistance[{1, 2, 3}, {5, 6, 7}] (* test built in function *)

myEuclideanDistance[{1, 2, 3, 4}, {5, 6, 7, 8}] (* test my function *)
EuclideanDistance[{1, 2, 3, 4}, {5, 6, 7, 8}] (* test built in function *)

myEuclideanDistance[{1, 2}, {5, 6, 7}] (* test my function *)
EuclideanDistance[{1, 2}, {5, 6, 7}] (* test built in function *)
```

4

4

$4\sqrt{2}$

$4\sqrt{2}$

$4\sqrt{3}$

$4\sqrt{3}$

8

8

MapThread::mptc : Incompatible dimensions of objects at positions
{2, 1} and {2, 2} of MapThread[(#2 – #1)^2 &, {{5, 6, 7}, {1, 2}}]; dimensions are 3 and 2. ≫

$\left\{\left\{\sqrt{5+\left((\#2-\#1)^2\ \&\right)},\ \sqrt{6+\left((\#2-\#1)^2\ \&\right)},\ \sqrt{7+\left((\#2-\#1)^2\ \&\right)}\right\},\ \left\{\sqrt{1+\left((\#2-\#1)^2\ \&\right)},\ \sqrt{2+\left((\#2-\#1)^2\ \&\right)}\right\}\right\}$

```
EuclideanDistance[{1, 2}, {5, 6, 7}]
```

Computational Exercise 9

```
seq = Array[0.8^# &, 21, 0]
```

```
{1., 0.8, 0.64, 0.512, 0.4096, 0.32768, 0.262144, 0.209715, 0.167772,
 0.134218, 0.107374, 0.0858993, 0.0687195, 0.0549756, 0.0439805,
 0.0351844, 0.0281475, 0.022518, 0.0180144, 0.0144115, 0.0115292}
```

```
ser = FoldList[Plus, seq]
```

```
{1., 1.8, 2.44, 2.952, 3.3616, 3.68928, 3.95142,
 4.16114, 4.32891, 4.46313, 4.5705, 4.6564, 4.72512, 4.7801,
 4.82408, 4.85926, 4.88741, 4.90993, 4.92794, 4.94235, 4.95388}
```

```
GraphicsRow[{ListPlot[ser, PlotLabel → "Sequence", Filling → Axis],
  ListPlot[seq, PlotLabel → "Series", Filling → Axis]}]
```