

# Reservoir

A **reservoir** is a natural or artificial lake, storage pond, or impoundment from a dam which is used to store water. Reservoirs may result from an avalanche, ice formation or a landslide (**natural reservoir**) or may be created in river valleys by the construction of a dam or may be built by excavation in the ground or by conventional construction techniques such as brickwork or cast concrete (**artificial reservoir**). Natural reservoirs often see fractures arising after a certain amount of time, caused by a great accumulation of water. Geological research has evidenced that since the last ice age several major floods were caused by this phenomenon. The major reason for constructing a reservoir usually is to generate green energy, where the great decline that is created in this way is used to drive a water turbine that powers an electric generator. In other cases, a reservoir is made as a regulator and a supply for irrigation or drinking water.



Lake Kariba is the world's largest artificial lake and reservoir by volume. It lies 1300 kilometers upstream from the Indian Ocean, along the border between Zambia and Zimbabwe. Lake Kariba was filled between 1958 and 1963 following the completion of the Kariba Dam at its northeastern end, flooding the Kariba Gorge on the Zambezi River. Lake Kariba is over 223 kilometers long and up to 40 kilometers in width. It covers an area of 5580 square kilometers and its storage capacity is an immense 185 cubic kilometers.

## Assignment

If geologists investigate a site for the construction of a new reservoir, they first draw a two-dimensional profile of the elevations and depressions in the landscape. Based on such a profile, they can compute the storage capacity of the reservoir. If the soil on the site primarily consists of coarser materials (e.g. sand and gravel) they must also take into account the some amount of water will penetrate into the soil.

Define a class `Reservoir` that can be used to investigate the storage capacity of a reservoir based on a given two-dimensional profile of the landscape. The objects of this class must at least support the following methods:

- An initialization method `__init__` that takes the profile of a landscape as its argument. This profile is represented as a list or tuple of positive integers. Each integer indicates the elevation of the landscape at the next sample point along the profile (we assume that sample points are equidistant).
- A method `__str__` that returns a string representation of the reservoir. This string

representation is formatted as a rectangular grid, with each column indicating the height at the corresponding sample point along the profile. The width of the grid is thus equal to the number of sample points along the profile. The number of hash symbols (#) at the bottom of each column corresponds to the height of the landscape at the sample point. On top of the stack of hash symbols, the column is filled with spaces. The number of lines in the string representation (and thus also the number of rows in the grid) is determined as the maximal height across all sample points along the profile. When the reservoir is filled (see below), the positions in the grid that are occupied by water are represented by a tilde (~) instead of a space.

- A method `fill` that fills the reservoir to its maximal water capacity. In doing so, it is assumed that the soil of the reservoir is considered impermeable, but that water can drain along the edges. An empty position in the grid representation of the reservoir (represented by a space in the string representation) is then occupied by water if that position is located in between two elevations (represented by hash symbols in the string representation) that are at least as high as the height of the position. The method must return the number of empty positions that are occupied by water when applying this procedure. After the method is called, the `__str__` method must return the string representation of the reservoir that is filled to its maximal capacity.
- A method `drain` that removes all water from the reservoir. The method must return the volume of water that was removed from the reservoir (counted as the number of positions in the string representation where a tilde was replaced by a space). After the method is called, the `__str__` method must return the string representation of the reservoir that is completely empty.
- A method `penetrate` that computes the volume of water that can penetrate into the soil. In doing so, it is assumed that the soil of the reservoir (where the landscape has height zero) is completely permeable. All water in the reservoir that touches the soil directly or indirectly (via water to the left, right or bottom) seeps away. The method must return the volume of water that seeps away from the reservoir when applying this procedure (counted as the number of positions in the grid representation of the reservoir containing water before calling the method where the water has seeped away after calling the method). After the method is called, the `__str__` method must return the string representation of the reservoir must represent positions where water has seeped away using a space instead of a tilde (~).

## Example

```
>>> profile = [4, 3, 2, 2, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 2, 5, 6, 5, 2, 2, 2, 3, 3, 3, 4, 5, 3, 2, 2]
>>> lake = Reservoir(profile)
>>> print(lake)
#
### #
#   ###  ##
##   ### #####
####  #####
##### #####
>>> lake.fill()
63
>>> print(lake)
#
###~::~::~~#
#~::~::~~###~::~::~~##
##~::~::~~###~::~::~#####
####~::~::~~#####
```

```
#####~::~~::~~#####
>>> lake.drain()
63
>>> print(lake)
#
### #
# ### ##
## ### #####
#### #####
##### #####
>>> lake.fill()
63
>>> print(lake)
#
###~::~~::~~#
#~::~~::~~#####~::~~::~~##
##~::~~::~~#####~::~~::~~#####
###~::~~::~~#####~::~~::~~#####
#####~::~~::~~#####~::~~::~~#####
>>> lake.penetrate()
47
>>> print(lake)
#
###~::~~::~~#
# ###~::~~::~~##
## ###~::~~::~~#####
#### #####
##### #####
>>> lake.fill()
47
>>> print(lake)
#
###~::~~::~~#
#~::~~::~~#####~::~~::~~##
##~::~~::~~#####~::~~::~~#####
###~::~~::~~#####~::~~::~~#####
#####~::~~::~~#####~::~~::~~#####
>>> lake.drain()
63
>>> print(lake)
#
### #
# ### ##
## ### #####
#### #####
##### #####
```

Een **stuwmeer** is een natuurlijk of kunstmatig meer dat ontstaat doordat de loop van een rivier wordt onderbroken. Dat kan bijvoorbeeld gebeuren door een sneeuwlawine, ijsvorming of een aardverschuiving (**natuurlijk stuwmeer**), of door een stuwdam in de rivier te plaatsen die het water tegenhoudt (**kunstmatig stuwmeer**). Bij natuurlijke stuwmeren ontstaan er na verloop van tijd vaak breuken door de grote ophoping van water. Geologisch onderzoek heeft uitgewezen dat er zich op die manier meerdere gigantische overstromingen hebben voorgedaan sinds de laatste ijstijd. De bouw van een stuwdam heeft meestal als doel om groene energie op te wekken, waarbij het groot verval dat op deze manier ontstaat gebruikt wordt om een waterturbine aan te drijven die een elektrische generator laat draaien. In andere gevallen wordt een stuwmeer aangelegd als regulator en voorraad voor irrigatie of drinkwater.



Het Karibameer is het grootste kunstmatige stuwmeer ter wereld, dat ontstaan is als gevolg van de bouw (1955-1959) van de Karibastuwdam in de Zambezi rivier, aan de grens van Zimbabwe en Zambia. Het meer ligt zo'n 385 kilometer stroomafwaarts van de Victoria watervallen, en heeft een lengte van 290 km en een maximale breedte van 32 km. De totale oppervlakte is 6000 km<sup>2</sup>.

## Opgave

Als geologen een site voor de aanleg van een nieuw stuwmeer onderzoeken, dan maken ze in de eerste plaats een twee-dimensionaal profiel op van de hoogtes en laagtes in het landschap. Op basis hiervan kunnen ze de watercapaciteit van het stuwmeer berekenen. Als de bodem op de site voornamelijk bestaat uit grovere materialen (bijvoorbeeld zand en grind), dan moet ook rekening gehouden worden met de hoeveelheid water die in de bodem zal dringen.

Definieer een klasse `Stuwmeer` waarmee de watercapaciteit van een stuwmeer kan onderzocht worden op basis van een gegeven twee-dimensionaal profiel van het landschap. De objecten van deze klasse moeten minstens de volgende methoden hebben:

- Een initialisatiemethode `__init__` waaraan een landschapsprofiel moet doorgegeven worden. Dit profiel bestaat uit een lijst of tuple van natuurlijke getallen. Elk getal geeft de hoogte van het landschap aan op een volgend meetpunt langs het profiel (we veronderstellen dat de meetpunten op gelijke afstand van elkaar liggen).
- Een methode `__str__` die een stringvoorstelling van het stuwmeer teruggeeft. Deze stringvoorstelling vormt een rechthoekig rooster waarvan elke kolom de hoogte aangeeft op het corresponderende meetpunt langs het landschapsprofiel. De breedte van het rooster is dus gelijk aan het aantal meetpunten langs het profiel. Elke kolom bevat onderaan evenveel hekjes (`#`) als de hoogte van het landschap op dat meetpunt. Daarboven wordt de kolom opgevuld met spaties. Het aantal regels in de stringvoorstelling (en dus ook het aantal rijen van het rooster) wordt bepaald door de maximale hoogte over alle meetpunten. Indien het stuwmeer wordt opgevuld (zie verder), dan moet een positie binnen het rooster weergegeven worden met een tilde (`~`) in plaats van een spatie als zich daar water bevindt.
- Een methode `vullen` waarmee het stuwmeer bijgevuld wordt tot zijn maximale watercapaciteit. Hierbij wordt verondersteld dat de bodem van het stuwmeer ondoordringbaar is, maar dat het water langs de randen wel kan weglopen. Een lege positie in de roostervoorstelling van het stuwmeer (voorgesteld door een spatie in de stringvoorstelling) wordt dan opgevuld met water, als die positie zich tussen twee hoogtes (voorgesteld door hekjes in de stringvoorstelling) bevindt die minstens even hoog zijn als de lege positie. De methode moet het aantal lege posities die op deze manier werden

opgevuld met water als resultaat teruggeven. Nadat de methode werd aangeroepen, moet de `__str__` methode de stringvoorstelling teruggeven van het stuwmeer dat is opgevuld tot zijn maximale watercapaciteit.

- Een methode `lozen` waarmee al het water uit het stuwmeer verwijderd wordt. De methode moet teruggeven hoeveel water er uit het meer verwijderd werd (geteld als het aantal posities in de stringvoorstelling waar de tilde werd vervangen door een spatie). Nadat de methode werd aangeroepen, moet de `__str__` methode de stringvoorstelling teruggeven van het stuwmeer dat volledig droog staat.
- Een methode `infiltreren` waarmee nagegaan wordt hoeveel water dat in het meer staat in de grond kan dringen. Hierbij wordt verondersteld dat de onderlaag van de bodem (daar waar het landschap hoogte nul heeft) volkomen doordringbaar is. Alle water die in het meer staat en die rechtstreeks of onrechtstreeks (via water links, rechts of onder) raakt aan de onderlaag van de bodem sijpelt weg. De methode moet teruggeven hoeveel water er op die manier wegsijpelt, geteld als het aantal posities in de roostervoorstelling van het stuwmeer die water bevatten vóór het aanroepen van de methode, en waar het water is weggesijpeld na het aanroepen van de methode. Nadat de methode werd aangeroepen, moet de `__str__` methode de posities waar water is weggesijpeld terug aangeven met een spatie in plaats van met een tilde (~).

## Voorbeeld

```
>>> profiel = [4, 3, 2, 2, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 2, 5, 6, 5, 2, 2, 2, 3, 3, 3, 4, 5, 3, 2, 2]
>>> meer = Stuwmeer(profiel)
>>> print(meer)
#
### #
# ### ##
## ### #####
#### #####
##### #####
>>> meer.vullen()
63
>>> print(meer)
#
###~~~~~#
#~~~~~###~~~~~##
##~~~~~###~~~~~#####
###~~~~~#####
#####~~~~~#####
>>> meer.lozen()
63
>>> print(meer)
#
### #
# ### ##
## ### #####
#### #####
##### #####
>>> meer.vullen()
63
>>> print(meer)
#
###~~~~~#
#~~~~~###~~~~~##
##~~~~~###~~~~~#####
```

```
###~::~~::~~::~~::~~::~~::~#####
#####~::~~::~~::~~::~~::~~::~#####
>>> meer.infiltreren()
47
>>> print(meer)
#
###~::~~::~~::~~::~~::~~::~#
#
###~::~~::~~::~~::~~::~~::~##
##
###~::~~::~#####
####
#####
#####
>>> meer.vullen()
47
>>> print(meer)
#
###~::~~::~~::~~::~~::~~::~#
#~::~~::~~::~~::~~::~~::~#####~::~~::~~::~~::~~::~~::~##
##~::~~::~~::~~::~~::~~::~#####~::~~::~~::~~::~~::~~::~#####
###~::~~::~~::~~::~~::~~::~#####
#####~::~~::~~::~~::~~::~~::~#####
>>> meer.lozen()
63
>>> print(meer)
#
### #
#
### ##
##
### #####
####
#####
#####
```