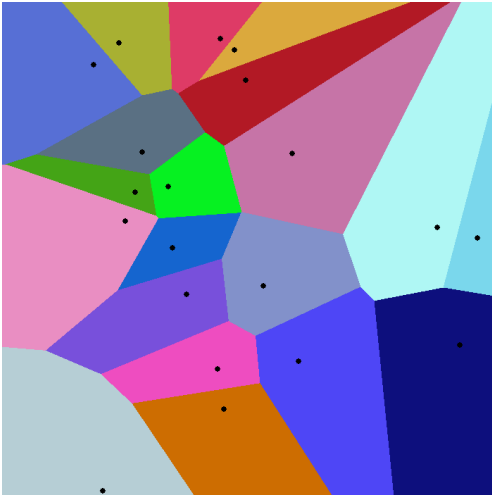


# Hospitals

In a certain region, there are a number of hospitals that are situated in a close range from one another. The competition between these hospitals is lethal, and that is to be taken literally. Patients that are transported by ambulance, are not always brought to the nearest hospital. Consequentially, valuable time is lost, sometimes with lethal consequences. The local authorities want to put an end to this, and have put out a new rule that obliges the paramedics to bring their patients to the nearest hospital. In order to do so, a map must be constructed dividing the region in zones with one hospital per zone.



Zone map that was constructed for 20 hospitals (situated in locations that were indicated with a black dot), with distances calculated based on the Euclidean distance.



Zone map that was constructed for 20 hospitals (the locations of which were indicated with a black dot), with distances calculated based on the Manhattan distance.

The authorities are asking you to make a division of a region, so that the nearest hospital on every point of a zone is always the only hospital situated in that zone. The individual hospitals, however, want to keep their sphere of influence as large as possible, and therefore, there is a lot of discussion about the way in which the distance between two points must be calculated. The table below contains a number of possible definitions the hospitals propose in order to calculate the distance between two points  $(x_1, y_1)$  en  $(x_2, y_2)$ .

name	formula
Euclidean distance	$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$

Manhattan distance  $d=|x_1-x_2| + |y_1-y_2|$

chessboard distance  $d=\max(|x_1-x_2|,|y_1-y_2|)$

## Assignment

Consider a rectangular region in which a number of hospitals are situated. We have divided this region in square cells that form a grid, so that each cell contains one hospital at the most. The rows and columns of the grid are respectively numbered from top to bottom and from left to right, starting from 0. The cells of the grid in which the hospital is situated, is marked with a (unique) uppercase letter and the remainder is marked with a full stop. The positions of the hospital can now be represented by a map of the following format:

```
.....B..
.....
...A.....
.....
.....D.
.....
.C.....
....E...
.....
```

The position of a cell in the grid is represented by a tuple  $(r, k)$ . Here,  $r$  is the row number and  $k$  is the column number for a cell in the grid. Asked:

- Write three functions `euclideanDistance`, `manhattanDistance` and `chessboardDistance` that correspond with the concepts of the same name for the word distance as it is defined in the above table. To each of these functions, two mandatory arguments must be passed. Every argument is a tuple of 2 integers that represent the  $x$  and  $y$  co-ordinate of a point in the surface. As a result, these functions must print the distance between the two given points as an integer, corresponding to the definitions of distance in the above table.
- Write a function `positions` to which the name of a text file must be passed. This text file must represent the positions of the hospitals in a certain region, in accordance with the format that was described above. The function must print a tuple of three elements, the first two of which respectively indicate the number of rows and columns of the rectangle grid in which the region was divided. The position of a hospital here corresponds with the position of a cell in the grid in which the hospital is situated.
- Write a function `nearest` with which the nearest hospital can be determined from a given point. To this function, two arguments must be passed to this function: the position of the cell in which the given point is situated and a dictionary that contains the positions of the hospitals in the regions. This dictionary must have the same format as the dictionaries that are printed by the function `positions` (as third element of the tuple). In order to determine the distance between a given point and a hospital, a distance function should be used to which two cell positions in the grid must be given and that prints an integer that represents the distance between these two cells. The function `euclideanDistance` must be used as a standard, but the function `nearest` also has a third optional parameter `distance` to which an alternative distance function can be given. The function must print the uppercase letter that is used to mark the hospital that is closest to the given point. If the given point is on an equal distance

from two or more hospitals, the upper case letter that alphabetically comes first, must be printed.

- Write a function `zones` to which the name of a text file must be given. This text file must represent the position of the hospitals in a certain region, according to the format that is described above. For every cell in the grid that is marked with a full stop, the function must determine the uppercase letter of the nearest hospital (cf. function `nearest`) and print the corresponding lowercase letter in the position of the full stop. By default, the result must be printed by the function. However, if a second file name is given to the function, the function must pass the result to a file with the given name. The function also has a third, optional parameter `distance`, that is used in the same way as the parameter with the same name in the function `nearest`.

## Example

In the example session below, we assume that the file [hospitals.txt](#) is situated in the current directory.

```
>>> euclideanDistance((0, 0), (3, 4))
5.0
>>> manhattanDistance((0, 0), (3, 4))
7.0
>>> chessboardDistance((0, 0), (3, 4))
4.0

>>> positions('hospitals.txt')
(9, 9, {(0, 6): 'B', (4, 7): 'D', (7, 5): 'E', (2, 3): 'A', (6, 1): 'C'})

>>> nearest((0, 0), {(0, 6): 'B', (4, 7): 'D', (7, 5): 'E', (2, 3): 'A', (6, 1): 'C'})
'A'
>>> nearest((8, 8), {(0, 6): 'B', (4, 7): 'D', (7, 5): 'E', (2, 3): 'A', (6, 1): 'C'}, distance=manhattanDistance)
'E'

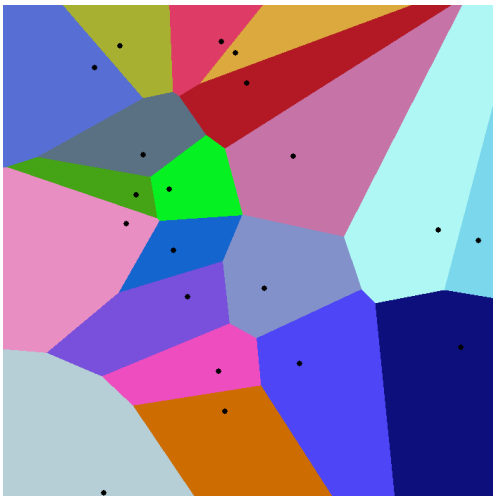
>>> zones('hospitals.txt')
aaaabbBbb
aaaaabbbb
aaaAaabdd
aaaaaaddd
ccaaddDd
cccceedd
cCcceedd
ccceEeee
ccceeeee

>>> zones('hospitals.txt', distance=manhattanDistance)
aaaabbBbb
aaaaabbbb
aaaAaabdd
aaaaaaddd
ccaaddDd
ccaedd
cCcceedd
ccceEeee
ccceeeee

>>> zones('hospitals.txt', distance=chessboardDistance)
aaaabBbb
aaaaabbbb
aaaAaabbb
```

```
aaaaaddd  
caaaaadDd  
ccccedddd  
cCcceeedd  
cccceEeed  
cccceeeee  
>>> zones('hospitals.txt', 'zones.txt')
```

In een bepaalde regio liggen er heel wat ziekenhuizen dicht bij elkaar. De concurrentie tussen deze ziekenhuizen is moordend, en dat valt letterlijk te nemen. Bij noodoproepen worden patiënten door een ziekenwagen immers niet altijd naar het dichtsbijzijnde ziekhuis gebracht. Hierdoor gaat kostbare tijd verloren, soms met dodelijke gevolgen. De plaatselijke overheid wil hier paal en perk aan stellen, en vaardigt een nieuwe regelgeving uit die ziekenwagens verplicht om hun patiënten altijd naar het dichtsbijzijnde ziekhuis te brengen. Hiervoor moet een plan opgesteld worden dat de regio opdeelt in zones waarin telkens één ziekenhuis gelegen is.



Zoneplan dat werd opgesteld voor 20 ziekenhuizen (gelegen op locaties aangeduid met een zwart punt), waarbij afstanden berekend werden op basis van de Euclidische afstand.



Zoneplan dat werd opgesteld voor 20 ziekenhuizen (gelegen op locaties aangeduid met een zwart punt), waarbij afstanden berekend werden op basis van de Manhattan-afstand.

De overheid vraagt je om een indeling van een regio te maken, zodat het dichtste ziekenhuis op elk punt in een zone telkens het enige ziekenhuis is dat in die zone gelegen is. De individuele ziekenhuizen willen echter hun invloedssfeer zo groot mogelijk houden, en daarom is er nog wat discussie over de manier waarop de afstand tussen twee punten moet berekend worden. Onderstaande tabel bevat een aantal mogelijke definities die de ziekenhuizen hebben voorgesteld om de afstand tussen twee punten  $(x_1, y_1)$  en  $(x_2, y_2)$  te berekenen.

**naam****formule**

Euclidische afstand  $d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$

Manhattan-afstand  $d = |x_1 - x_2| + |y_1 - y_2|$

schaakbord-afstand  $d = \max(|x_1 - x_2|, |y_1 - y_2|)$

**Opgave**

Beschouw een rechthoekige regio waarin een aantal ziekenhuizen gelegen zijn. We hebben deze regio opgedeeld in vierkante cellen die een rooster vormen, zodat elke cel hoogstens één ziekenhuis bevat. De rijen en kolommen van het rooster worden respectievelijk van boven naar onder en van links naar rechts genummerd vanaf 0. De cellen van het rooster waarin een ziekenhuis gelegen is, markeren we met een (unieke) hoofdletter en de overige cellen markeren we met een punt. Daardoor kan de ligging van de ziekenhuizen in de regio voorgesteld worden door een kaart die de volgende vorm heeft:

```
.....B..
.....
...A....
.....
.....D.
.....
.C.....
....E...
.....
```

De positie van een cel in het rooster wordt voorgesteld door een tuple  $(r, k)$ . Hierbij stelt  $r$  het rijnummer en  $k$  het kolomnummer voor van de cel in het rooster. Gevraagd wordt:

- Schrijf drie functies `euclidischeAfstand`, `manhattanAfstand` en `schaakbordAfstand` die corresponderen met de gelijknamige concepten voor het begrip afstand zoals die in bovenstaande tabel gedefinieerd werden. Aan elk van deze functies moeten verplicht twee argumenten doorgegeven worden. Elk argument is een tuple van 2 reële getallen die de  $x$ - en de  $y$ -coördinaat van een punt in het vlak voorstellen. Als resultaat geven deze functies telkens de afstand tussen de twee gegeven punten terug als een reëel getal, overeenkomstig de corresponderende definitie van afstand.
- Schrijf een functie `posities` waaraan de naam van een tekstbestand moet doorgegeven worden. Dit tekstbestand moet de ligging van de ziekenhuizen in een bepaalde regio voorgestellen, volgens het formaat dat hierboven werd beschreven. De functie moet een tuple met drie elementen teruggeven, waarvan de eerste twee respectievelijk het aantal rijen en het aantal kolommen van het rechthoekige rooster aangeven waarin de regio werd opgedeeld. Het derde element is een dictionary, die de positie van elk ziekenhuis in de regio afbeeldt op de hoofdletter waarmee het ziekenhuis in het rooster gemarkeerd werd. De positie van een ziekenhuis komt hierbij overeen met de positie van de cel in het rooster waarin het ziekenhuis gelegen is.
- Schrijf een functie `dichtste` waarmee het dichtste ziekenhuis kan bepaald worden vanaf een gegeven punt. Aan deze functie moeten twee argumenten doorgegeven worden: de positie

van de cel waarin het gegeven punt gelegen is en een dictionary die de posities van de ziekenhuizen in de regio bevat. Deze dictionary moet dezelfde vorm hebben als de dictionaries die door de functie `posities` wordt teruggegeven (als derde element van het tuple). Om de afstand tussen het gegeven punt en een ziekenhuis te bepalen, moet een afstandsfunctie gebruikt worden waaraan twee celposities in het rooster moeten doorgegeven worden en die een reëel getal teruggeeft dat de afstand tussen deze twee cellen voorstelt. Standaard moet hiervoor de functie `euclidischeAfstand` gebruikt worden, maar de functie `dichtste` heeft ook nog een derde optionele parameter `afstand` waaraan een alternatieve afstandsfunctie kan doorgegeven worden. De functie moet de hoofdletter teruggeven waarmee het ziekenhuis gemarkeerd is dat dichtst bij het gegeven punt gelegen is. Indien het gegeven punt op gelijke dichtste afstand ligt van twee of meer ziekenhuizen, dan moet de alfabetisch eerst gerangschikte hoofdletter teruggegeven worden waarmee deze ziekenhuizen gemarkeerd zijn.

- Schrijf een functie `zones` waaraan de naam van een tekstbestand moet doorgegeven worden. Dit tekstbestand moet de ligging van de ziekenhuizen in een bepaalde regio voorgestellen, volgens het formaat dat hierboven werd beschreven. Voor elke cel in het rooster die gemarkeerd wordt door een punt, moet de functie de hoofdletter bepalen van het dichtstbijzijnde ziekenhuis (cf. functie `dichtste`) en de corresponderende kleine letter invullen in plaats van het punt. Standaard moet het resultaat door de functie uitgeschreven worden. Als aan de functie echter ook nog een tweede bestandsnaam doorgegeven wordt, dan moet de functie het resultaat wegschrijven naar een bestand met de opgegeven naam. De functie heeft ook nog een derde optionele parameter `afstand`, die op dezelfde manier gebruikt wordt als de gelijknamige parameter van de functie `dichtste`.

## Voorbeeld

Bij onderstaande voorbeeldsessie gaan we ervan uit dat het bestand [ziekenhuizen.txt](#) zich in de huidige directory bevindt.

```
>>> euclidischeAfstand((0, 0), (3, 4))
5.0
>>> manhattanAfstand((0, 0), (3, 4))
7.0
>>> schaakbordAfstand((0, 0), (3, 4))
4.0

>>> posities('ziekenhuizen.txt')
(9, 9, {(0, 6): 'B', (4, 7): 'D', (7, 5): 'E', (2, 3): 'A', (6, 1): 'C'})

>>> dichtste((0, 0), {(0, 6): 'B', (4, 7): 'D', (7, 5): 'E', (2, 3): 'A', (6, 1): 'C'})
'A'
>>> dichtste((8, 8), {(0, 6): 'B', (4, 7): 'D', (7, 5): 'E', (2, 3): 'A', (6, 1): 'C'}, afstand=manhattanAfstand)
'E'

>>> zones('ziekenhuizen.txt')
aaaabbBbb
aaaaabbbb
aaaAaabdd
aaaaaaddd
ccaaaddDd
cccceedd
cCcceedd
ccceeEeee
```

```
ccccceeee
>>> zones('ziekenhuizen.txt', afstand=manhattanAfstand)
aaaabbBbb
aaaaabbbb
aaaAaabdd
aaaaaaddd
ccaaaddDd
cccaeeddd
cCcceeedd
ccceeEeee
ccccceeee
>>> zones('ziekenhuizen.txt', afstand=schaakbordAfstand)
aaaabBbb
aaaaabbbb
aaaAabbbb
aaaaaaddd
caaaaadDd
cccceddd
cCcceeedd
cccceEeed
ccccceeee
>>> zones('ziekenhuizen.txt', 'zones.txt')
```