Airports

The text file airports.txt contains information on all airports of the US. The information of each airport is on a separate line and consists of the following five fields of information that are separated by a tab:

- code: unique identification code for the airport
- latitude: latitude on which the airport is located (expressed in decimal degrees)
- longitude: longitude on which the airport is located (expressed in decimal degrees)
- city: name of the city in which the airport is located
- state: (abbreviated) name of the state in which the airport is located

The first line of the file is a header that contains the names of the different columns.

Assignment

- Write a function

  readairports(filename)

  that takes the location of the text file airports.txt (or a file with a similar layout) as an argument. This function must return a dictionary, where information about the airports can be retrieved quickly. The unique identifier of the airports is used as a key in this dictionary, and the associated value is a tuple consisting of the four information fields (latitude, longitude, city, state). Here, the latitude and longitude are to be included as real numbers in the tuple, and the city and country in which the airport is situated as strings.
- Write a function

  distance(code1, code2, airports)

  that returns the distance between any two airports (expressed in kilometres). The unique identifiers of two airports should be passed as an argument to this function along with a dictionary that contains the information about the airports (in the format as it is returned by the readairports function). To calculate the distance between two airports, you use the following formula $$ R\cdot \arctan\left(\frac{\sqrt{(\cos b_2\sin (l_1-l_2))^2 + (\cos b_1\sin b_2-\sin b_1\cos b_2 \cos(l_1-l_2))^2}}{\sin b_1\sin b_2+\cos b_1\cos b_2\cos(l_1-l_2)}\right) $$ This is actually a formula that calculates distance between two points on a sphere with a radius R. We hereby represent the Earth as a sphere with radius $R = 6372{,}795$ km. Furthermore, b_1 en b_2 (resp. l_1 and l_2) represent the latitude (resp. longitude) of the two points on the sphere, expressed in radians. For the conversion of degrees to radians you should use the value pi from the math module of the Python Standard Library.
- Write a function

  stopover(code1, code2, airports[, scope=4000])

  which can be used to determine the optimal flight schedule with a single stopover (for refueling), in case one wants to fly between two points of which the distance is greater than the maximum range of the aircraft. The optimal flight schedule to fly from point A to point B has a stopover in an airport at point C and where the distance from A to C and from C to B always lies within the range of the aircraft, and for which the total flight

distance is as small as possible. This function must carry the same three arguments as are passed to the `distance` function. An optional fourth argument can also be passed, which indicates the maximum range of the aircraft (in kilometres). By default, 4000 km is taken as maximum range. The function should return the unique identifier of the airport that ensures an optimal flight schedule as a result. The function should return a value of `None`, if the two specified airports are within the range of the aircraft (if it can be flown directly, in other words), or if a flight schedule with a single stopover is not possible.

Example

In the following interactive Python session we assume that the file airports.txt is located in the current directory.

```
>>> airports = readairports('airports.txt')
>>> airports
{'AGN': (57.83, 134.97, 'Angoon', 'AK'), ...}
>>> airports['ADK']
(51.88, 176.65, 'Adak', 'AK')
>>> airports['DCA']
(38.85, 77.04, 'Washington/Natl', 'DC')
>>> airports['4OM']
(48.42, 119.53, 'Omak', 'WA')
>>> distance('P60', 'MSN', airports)
1694.54554995
>>> distance('ADK', 'DCA', airports)
7295.50355678
>>> stopover('ADK', 'DCA', airports, 4000)
'4OM'
```

Het tekstbestand luchthavens.txt bevat informatie over alle luchthavens van de Verenigde Staten. De informatie van elke luchthaven staat op een afzonderlijke regel, en bestaat uit de volgende vijf informatievelden die van elkaar worden gescheiden door een tab:

- `code`: unieke identificatiecode voor de luchthaven
- `latitude`: breedtegraad waarop de luchthaven gelegen is (uitgedrukt in decimale graden)
- `longitude`: lengtegraad waarop de luchthaven gelegen is (uitgedrukt in decimale graden)
- `city`: naam van de stad waarin de luchthaven gelegen is
- `state`: (afgekorte) naam van de staat waarin de luchthaven gelegen is

De eerste regel van het bestand is een hoofding die de namen van de verschillende kolommen bevat.

Opgave

- Schrijf een functie

  `leesLuchthavens(bestandsnaam)`

  waaraan de locatie van het tekstbestand luchthavens.txt (of een bestand met een gelijkaardige opmaak) moet doorgegeven worden. Deze functie moet als resultaat een dictionary teruggeven, waarin informatie over de luchthavens snel kan opgezocht worden. De unieke identificatiecode van de luchthavens wordt in deze dictionary als sleutel gebruikt, en de geassocieerde waarde is een tuple dat bestaat uit de vier informatievelden

(latitude, longitude, city, state). Hierbij moeten de breedte- en lengtegraad als reële getallen opgenomen worden in de tuple, en de stad en staat waarin de luchthaven gelegen is als strings.

- Schrijf een functie

  afstand(code1, code2, luchthavens)

  die de afstand tussen twee gegeven luchthavens teruggeeft (uitgedrukt in kilometer). Aan deze functie moeten de unieke identificatiecodes van twee luchthavens als argument doorgegeven worden, samen met een dictionary die de informatie over de luchthavens bevat (in het formaat zoals deze door de functie leesLuchthavens wordt teruggegeven).
  Om de afstand tussen twee luchthavens te berekenen, maak je gebruik van onderstaande formule \[R\cdot \arctan\left(\frac{\sqrt{(\cos b_2\sin (l_1-l_2))^2 + (\cos b_1\sin b_2-\sin b_1\cos b_2 \cos(l_1-l_2))^2}}{\sin b_1\sin b_2+\cos b_1\cos b_2\cos(l_1-l_2)}\right) \] Dit is eigenlijk een formule die de afstand tussen twee punten op een bol met straal R berekent. We stellen de Aarde hierbij dus benaderend voor als een bol, met straal $R = 6372{,}795$ km. Voorts stellen b_1 en b_2 (resp. l_1 en l_2) de breedteligging (resp. lengteligging) voor van de twee punten op de bol, uitgedrukt in radialen. Voor de omzetting van graden naar radialen moet je gebruik maken van de waarde pi uit de module math van de Python Standard Library.

- Schrijf een functie

  tussenlanding(code1, code2, luchthavens[, reikwijdte=4000])

  die gebruikt kan worden om het optimale vluchtschema met één enkele tussenlanding (voor een extra tankbeurt) te bepalen, in het geval men een vlucht wil maken tussen twee punten waarvan de afstand groter is dan de maximale reikwijdte van het vliegtuig. Het optimale vluchtschema om van punt A naar punt B te vliegen, heeft een tussenlanding in een luchthaven die op punt C gelegen is en waarvoor de afstand om van A naar C te vliegen en van C naar B te vliegen telkens binnen het bereik van het vliegtuig ligt, en waarvoor de totale vliegafstand zo klein mogelijk is. Aan deze functie moeten verplicht dezelfde drie argumenten als voor de functie afstand doorgegeven worden. Optioneel kan ook nog een vierde argument doorgegeven worden, dat de maximale reikwijdte van het vliegtuig (uitgedrukt in kilometer) aangeeft. Standaard wordt als maximale reikwijdte 4.000 km genomen. De functie moet als resultaat de unieke identificatiecode van de luchthaven teruggeven die zorgt voor een optimaal vluchtschema. De functie moet de waarde None teruggeven indien de twee opgegeven luchthavens binnen het bereik van het vliegtuig liggen (als er met andere woorden rechtstreeks kan gevlogen worden), of als een vluchtschema met één enkele tussenlanding niet mogelijk is.

Voorbeeld

In onderstaande interactieve Python sessie gaan we ervan uit dat het tekstbestand luchthavens.txt zich in de huidige directory bevindt.

```
>>> luchthavens = leesLuchthavens('luchthavens.txt')
>>> luchthavens
{'AGN': (57.83, 134.97, 'Angoon', 'AK'), ...}
>>> luchthavens['ADK']
(51.88, 176.65, 'Adak', 'AK')
>>> luchthavens['DCA']
(38.85, 77.04, 'Washington/Natl', 'DC')
```

```
>>> luchthavens['4OM']
(48.42, 119.53, 'Omak', 'WA')
>>> afstand('P60', 'MSN', luchthavens)
1694.54554995
>>> afstand('ADK', 'DCA', luchthavens)
7295.50355678
>>> tussenlanding('ADK', 'DCA', luchthavens, 4000)
'4OM'
```