

1 Is one Interval in some array of Intervals?

Suppose we have an array of Intervals (.1,.5), (.2,.7), and (.8,.9). If we ask whether an Interval (.3,.4) “is in” the array of Intervals, we will answer “yes, yes, no” since

```
(.3,.4) is in (.1,.5),
(.3,.4) is in (.2,.7), but
(.3,.4) is not in (.8,.9).
```

Recall that class `Interval` has an `isIn` method that would be useful here. Implement the following function (independent function, not an instance method):

```
function tf = isInRange(inter, interArray)
% inter: an Interval
% interArray: 1-d array of Intervals
% tf: a logical vector (1s and 0s) the same length as interArray where
%     tf(k)=1 if inter "is in" interArray(k); otherwise tf(k)=0.
```

Later using MATLAB, test your function by typing the following in the *Command Window*:

```
a= intervalArray(4); % Create a length 4 array of Intervals using the
                    %   function implemented in lecture.
b= Interval(.3,.5)
v= isInRange(b,a)   % Why isn't the function call something.isInRange(...) ?
                    % Answer: isInRange isn't an instance method; it's its own function.
```

2 Analyze class LocalWeather

A partially completed classdef for `LocalWeather` is given on the next page. Read the properties and the constructor of class `LocalWeather`; ask questions if there is anything that you do not understand. Answer the following questions and then complete methods `getAnnualPrecip` and `getMonthlyAveTemps`.

4.1 By reading class `LocalWeather` above and the file `ithacaWeather.txt`, answer the following questions.

```
ithaca= LocalWeather('ithacaWeather.txt')
disp(ithaca.city)      % What is the output? -----

disp(ithaca.precip)    % What is the output? -----

disp(ithaca.precip(11)) % What is the output? -----

a= ithaca.temps        % What is the type of a? -----

b= ithaca.temps(11)    % What is the type of a? -----

disp(ithaca.temps(11).left) % What is displayed? What is it? -----
```

4.2 Implement function `getAnnualPrecip`, which calculates and returns the total annual precipitation. If any month’s precipitation data is missing, the returned value should be `NaN`, a value in Matlab of type `double` that indicates that a value is not-a-number.

4.3 Implement function `getMonthlyAveTemps` which returns the vector (length 12) of monthly average temperatures. Calculate a month’s average temperature as the average between the month’s high and low temperatures. If a month is missing temperature data, its average temperature should be set to `NaN`. The built-in function `isnan` can be used to check whether a variable stores the value `NaN`: `isnan(x)` returns true (1) if `x` is `NaN` and false (0) otherwise.

Later, using MATLAB, type in your method bodies and test the updated class by creating a `LocalWeather` object and calling its methods.

```

classdef LocalWeather < handle
% Weather data (monthly low, high, and precip) for a city from a standard
% city weather data file.

properties
    city= ''; % City name string
    temps= Interval.empty();
        % array of 12 Intervals, each (monthly low, monthly high)
    precip % numeric vector of length 12, each monthly precipitation
end

methods
function lw = LocalWeather(fname)
    if nargin==1
        fid= fopen(fname, 'r');
        % Get city name
        s= fgetl(fid);
        lw.city= s(3:length(s));
        % Read pass headers lines (next 3 lines)
        for k=1:3
            s= fgetl(fid);
        end
        % Read monthly data (next 12 lines)
        for k= 1:12
            s= fgetl(fid);
            lw.temps(k)= Interval(str2double(s(4:8)), ...
                str2double(s(12:16)));
            lw.precip(k)= str2double(s(20:24));
        end
        fclose(fid);
    end
end

function showMonthData(self, m)
% Show data for month m. m is an integer and 1<=m<=12.
mo= {'Jan', 'Feb', 'Mar', 'Apr', 'May', 'June', ...
    'July', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec'};
fprintf('%s Data\n', mo{m})
fprintf('Temperature range: ')
disp(self.temps(m))
fprintf('Average precipitation: %.2f\n', self.precip(m))
end

function p = getAnnualPrecip(self)
% If any month is missing precip data, display a warning message
% and p is NaN. Otherwise p is annual precipitation.
p=0;
end

function tempVec = getMonthlyAveTemps(self)
% tempVec is length 12 vector of monthly average temperatures.
% tempVec(m) is average between month m's high and low temps.
% If a month is missing either high or low temp (NaN), then that
% month's average is also NaN.
tempVec= zeros(1,12);
end

function showCityName(self)
    disp(self.city)
end
end %methods
end %classdef

```