

Guohao Lin
NEPTUNCODE
IW3XV9
Group 1

2. assignment/3. task

April 2020

Task

The results of the National Angler's Championship is stored in a text file. Each line of the file contains the identifier of the participant and the championship (strings without whitespace), and the list of the caught fish, which are stored as pairs: (the kind of the fish, the size of the fish). The kind of the fish is a string without whitespace, its size is a natural number. The data in a line are separated by whitespace. The lines of the text file are sorted according to the name of the championship. You can assume that the text file is correct. An example for a line of the text file:

James BigLakeChampionship Tuna 50 Salmon 20 Sardine 5 Tuna 100

List all the championships where tuna has been caught.

Plan of the main program

Specification:

$A = (x:infile(S), l: L, e: Championship) \text{ Championship} = \text{rec}(ID:String, caught:L)$
 $Pre = (x = x' \wedge x \nearrow) \text{ (} x \text{ is sorted as pairs and sorted according to the name of the championship)}$

New specification: $\text{Championship} = \text{rec}(ID:String, caught:L)$

$A = (t: \text{Enor}(\text{Championship}), l: L, e: \text{Championship})$

$Pre = (t = t')$

$Post = (l, e) = (\sum_{e \in t} \text{print}(e.ID))$

$Cond(e): e.caught = true$

Summation:

$t: \text{Enor}(E)$	$\sim t: \text{Enor}(\text{Championship}), \text{ where } (ID, caught) := t.current(), (ID, caught) \text{ is denoted by } e.$
$f(e)$	$\sim e.caught = true \rightarrow \text{print}(e.ID)$ $e.caught = false \rightarrow \text{SKIP}$
S	$\sim e.ID$
$H, +, 0$	$\text{Championship}^*, \oplus, \text{Print}$

Algorithm:

$l := false;$	
$t.first();$	
$\neg t.end()$	
$\backslash t.current().caught = true \ /$	
$l := true$	$t.next()$
$\text{print}(t.current().ID);$	
$t.next();$	

Enumerator of Championship

t : Enor(Championship)

Championship = rec(ID:String, caught:L)

Championship*	first()	next()	current() : Championship	end() :L	read()	Judge(dx):L
x : infile(S) dx:String sx:Status curr : Championship end : L name:S	sx,dx,x: read() next()	see below	return curr	return end	See below information	Determine whether a line (dx) in the file contains "Tuna"

The name is a string which to store every second word in each line(Championship names).
This will make it easier for us to compare the championship name.

Use the method read () which in the t:Enor(Championship) to read line by line from the file to determine whether lines with the same “championship” (second element) contain "Tuna".
When the second element “championship” in the row does not match the “championship” in the previous row. Stop the read () operation. Process data into Championship type.

The read() method in the t:Enor(Championship) should read one line from the file and return the string when the file is not empty.If the file is empty , make the Status as abnorm, otherwise Status is norm. This method also need to pick out the second element of each line and store it to the “name”

Championship = rec(ID:String, caught:L)

next() method: $S = (x:\text{infile}(S), dx:s, sx:\text{Status}, curr:\text{Championship}, end:L)$ $Pre = (x = x' \wedge x \nearrow \wedge dx = dx' \wedge sx = sx')$ $Post = (pre \wedge (curr.ID = \text{"Second word of } dx\text{"}) \wedge (\neg end \rightarrow curr.caught = \bigvee_{dx \in (dx', x')} Judge(dx)) \wedge$ $end = (sx = \text{abnorm})$ $Cond(dx): \text{"Second word of } dx\text{" (name) = curr.ID}$ *dx means one line from the input file.*

Summation(OR's ing)

$t : \text{Enor}(\text{Championship}) \sim x:\text{infile}(S) \text{ (sx.dx.x:read)}$
 $\text{cond: "Second word of dx"(name) = curr.ID}$
 $f(e) \sim \text{Judge}(dx)$
 $S \sim \text{curr.caught}$
 $H, +, 0 \sim L, \vee, \text{false}$

Algorithm:

end:= (sx=abnorm)	
end()	
T	F
SKIP	curr.caught:=false
	curr.ID:=name
	(sx=norm)&&(name= curr.ID)
	curr.caught:=curr.caught \vee Judge(dx)
	sx,dx,x::read

*name means the
second word of the string(dx)*

Judge(dx) method:

$S = (\text{onefish}:\text{Fish}, dx:S, l:L) \text{ Fish} = \text{rec}(\text{fishname}:S, \text{size}:N) \quad \text{onefish represents every possible fish}$
in dx

$\text{Pre} = (dx = dx', l = \text{false})$

$\text{Post} = (\text{pre} \wedge l = \bigvee_{\text{onefish} \in dx'} \text{cond}(\text{onefish}) := (\text{onefish.fishname} = \text{"Tuna"}))$

*As long as there is the word "Tuna" in a
onefish.fishname, l is true.*

Summation(OR's ing)

$tt:\text{Enor}(\text{Fish}) \sim dx:\text{String} \text{ (sx._dx.x:read)} \quad _dx \text{ now should be Fish.}$
 $\text{Cond}(_dx) = (_dx.fishname = \text{"Tuna"})$
 $f(e) \sim \text{Cond}(_dx) = (_dx.fishname = \text{"Tuna"})$
 $S \sim l$
 $H, +, 0 \sim L, \vee, \text{false}$

Algorithm:

l:=false
fisher:=tt.read();
name:=tt.read()
onefish.fishname:=tt.read(); onefish.size:=tt.read()&&tt.end \neq true
l=(l \vee onefish.fishname = "Tuna")
return l

*The "fisher" is a string which
represents the name of the angle in
one line(first word of dx)*

tt: Enor(Fish)

Fish = rec(fishname:S,size:N)

Fish*	end() :L	read()
x : infile(S) _dx:String sx:Status _curr : Fish* end : L	return end	Read one word from the specified line

The read () method will return the words in the given string one by one. If words can be read, then make the sx as norm , otherwise make the sx as abnorm.

Testing plan

Two algorithmic patterns are used in the solution: Summation and Summation (OR's ing) .

- 1) Empty file
- 2) Two championships in two lines, the first one meets the condition.
- 3) Two championships in two lines and all of them meet the condition.
- 4) Two championships in three lines, the first two line have the same championship name, only first line meets the condition.
- 5) Two championships in three lines, the first two line have the same championship name, only first two lines meet the condition.
- 6) Two championships in three lines, the first two line have the same championship name, and all of them meet the condition.
- 7) Two championships in three lines, the first two line have the same championship name, only last line meet the condition.
- 8) Two championships in three lines, the first two line have the same championship name, only second line meets the condition.
- 9) Two championships in three lines, the first two line have the same championship name, no lines meet the condition.
- 10) Two championships in three lines, the first two line have the same championship name, only the first and the last line meet the condition.
- 11) Three championships in three lines, every line meets the condition.
- 12) 10 different championships in different lines, every of them meet the condition.
- 13) 10 different championships in different lines, every of them meet the condition except the first line.
- 14) One championship in 10 different lines, every of them meet the condition.
- 15) One championship in one line. The line contains a total of 3 kinds of fish, Tuna on the first.
- 16) One championship in one line. The line contains a total of 3 kinds of fish, Tuna on the second.
- 17) One championship in one line. The line contains a total of 3 kinds of fish, Tuna on the third.