

Program



CONTENTS

- 1 Loop
- 2 Dynamic
- 3 Subprogram
- 4 Parallel

01

Loop

Calculate the total salary of the company's employees (sum function)

	A	B
1	=demo.query("select * from EMPLOYEE")	
2	0	
3	for A1.len()	>A2=A2+A1(A3).SALARY

A1, A2 results :

Index	EID	NAME	SURNAME	GENDER	STATE	BIRTHDAY	HIREDATE	DEPT	SALARY
1	1	Rebecca	Moore	F	California	1974-11-20	2005-03-11	R&D	7000
2	2	Ashley	Wilson	F	New York	1980-07-19	2008-03-16	Finance	11000
3	3	Rachel	Johnson	F	New Mexico	1970-12-17	2010-12-01	Sales	9000
4	4	Emily	Smith	F	Texas	1985-03-07	2006-08-15	HR	7000
5	5	Ashley	Smith	F	Texas	1975-05-13	2004-07-30	R&D	16000
6	6	Matthew	Johnson	M	California	1984-07-07	2005-07-07	Sales	11000
7	7	Alexis	Smith	F	Illinois	1972-08-16	2002-08-16	Sales	9000
8	8	Megan	Wilson	F	California	1979-04-19	1984-04-19	Marketing	11000
9	9	Victoria	Davis	F	Texas	1983-12-07	2009-12-07	HR	3000
10	10	Ryan	Johnson	M	Pennsylva...	1976-03-12	2006-03-12	R&D	13000

Value
3697500

Calculate the total salary of the company's employees (sum function)

	A	B
1	=demo.query("select * from EMPLOYEE")	
2	0	
3	for A1	>A2=A2+A3.SALARY

A1, A2 results:

Index	EID	NAME	SURNAME	GENDER	STATE	BIRTHDAY	HIREDATE	DEPT	SALARY
1	1	Rebecca	Moore	F	California	1974-11-20	2005-03-11	R&D	7000
2	2	Ashley	Wilson	F	New York	1980-07-19	2008-03-16	Finance	11000
3	3	Rachel	Johnson	F	New Mexico	1970-12-17	2010-12-01	Sales	9000
4	4	Emily	Smith	F	Texas	1985-03-07	2006-08-15	HR	7000
5	5	Ashley	Smith	F	Texas	1975-05-13	2004-07-30	R&D	16000
6	6	Matthew	Johnson	M	California	1984-07-07	2005-07-07	Sales	11000
7	7	Alexis	Smith	F	Illinois	1972-08-16	2002-08-16	Sales	9000
8	8	Megan	Wilson	F	California	1979-04-19	1984-04-19	Marketing	11000
9	9	Victoria	Davis	F	Texas	1983-12-07	2009-12-07	HR	3000
10	10	Ryan	Johnson	M	Pennsylva...	1976-03-12	2006-03-12	R&D	13000

Value
3697500

for conditions

Define customers whose sales reach half of total sales as "big customers". Find out these big customers.

	A	B
1	=file("E:\\txt\\Contract.txt").import@t()	
2	=A1.groups(Client;sum(Amount):S_amount).sort(-S_amount)	
3	=A1.sum(Amount)	
4	0	0
5	for A4<A3/2	>B4=B4+1
6		>A4=A4+A2(B4).S_amount
7		>A8=A8 A2(B4).Client
8		
9	=A2.iterate((A10=A10 ~.Client,~~+S_amount),0,~~>=A3/2)	

A8, A10

Index	Member
1	QUICK
2	SAVEA
3	ERNSH
4	HUNGO
5	RATTC
6	HANAR
7	FOLKO
8	MEREP
9	KOENE
10	QUEEN
11	WHITC
12	FRANK
13	BERGS

A9

Value
701553.5099999999

A1

Index	ContractNo	ActualSale	SellDate	Product	Quantity	Amount	Client	ApplyArea	ApplyMethod
1	10961	8	1997-03-1...	52	6	1122.0	QUEEN	EastChina	SELF_USE
2	10962	8	1997-03-1...	7	45	3584.0	QUICK	SouthChina	SELF_USE
3	10963	9	1997-03-1...	60	2	68.0	FURIB	NorthChina	RESELL
4	10964	3	1997-03-2...	18	6	2052.5	SPECD	SouthChina	RESELL
5	10965	6	1997-03-2...	51	16	848.0	OLDWO	EastChina	SELF_USE

A2

Index	Client	S_amount
1	QUICK	117483.39
2	SAVEA	115673.39
3	ERNSH	113236.68
4	HUNGO	57317.39
5	RATTC	52245.89999999...

A3

Value
1354510.0900000012

Stop looking for random employees if more than 30 or three are in New York State.

	A	B	C
1	=demo.query("select * from EMPLOYEE").sort(rand())		
2	[]	0	
3	for	=A1(A3)	
4		if B3.STATE=="New York"	>B2=B2+1
5		>A2=A2 B3.NAME	
6		if A2.len()>30	break
7		if B2>3	break

A2 result

5	Elizabeth
6	Olivia
7	Taylor
8	Elizabeth
9	Sarah
10	Sarah
11	Lauren
12	Jacob
13	Elizabeth
14	Antony
15	Elizabeth

Find out "Narcissus Number" within 1000

	A	B	C	D	E
1	=to(0,9)				
2	for A1	if A2==0	next		
3		=A2*A2*A2			
4		for A1	=B4*B4*B4		
5			if B3+C4>(A2+1)*100	break	
6			if C4>(A2+1)*100	break	
7			for A1	=C7*C7*C7	
8			if D7>(A2+1)*100		break C7
9			if D7+B3>(A2+1)*100		break
10			if D7+C4>(A2+1)*100		break
11			if B3+C4+D7>(A2+1)*100		break
12			=A2*100+B4*10+C7		
13			if B3+C4+D7==D12		>A14=A14 D12
14					

A14 result

1	153
2	370
3	371
4	407

02

Dynamic



Generate part of an expression dynamically through macros

Dynamic

In a game, players draw three random prizes, two for scores, one for operators, and the final score is the result calculated from the first score and the second score.

	A	B
1	=rand(100)+100	/First score
2	=rand(10)	/Second score
3	[+,-,*,/]	/Operator
4	=A3(rand(4)+1)	/The extracted operator
5	=A1\${A4}A2	/Final score

A1, A2, A4, A5 results:

Value
154

Value
3

Value
/

Value
51.333333333333336

A1-A5 recorded the population of five states. B1-B5 was the abbreviation of each state. Find the first state with a population of more than 5,000,000.

	A	B
1	4779736	AL
2	710231	AK
3	6392017	AZ
4	2915918	AR
5	37253956	CA
6	=[A1:A5].pselect(~>5000000)	
7	=B\${string(A6)}	

A6, A7 results:

Value
3

Value
AZ

Calculate the non-repetitive values of non-numeric fields

	A	B
1	=demo.query("select * from EMPLOYEE")	
2	=A1.fname()(A1(1).array().(ifnumber(~)).pos@a(false))	
3	for A2	=@[A1.id({A3})]

A1, A2 results:

Index	EID	NAME	SURNAME	GENDER	STATE	BIRTHDAY	HIREDATE	DEPT	SALARY
1	1	Rebecca	Moore	F	California	1974-11-20	2005-03-11	R&D	7000
2	2	Ashley	Wilson	F	New York	1980-07-19	2008-03-16	Finance	11000
3	3	Rachel	Johnson	F	New Mexico	1970-12-17	2010-12-01	Sales	9000
4	4	Emily	Smith	F	Texas	1985-03-07	2006-08-15	HR	7000
5	5	Ashley	Smith	F	Texas	1975-05-13	2004-07-30	R&D	16000

Index	Member
1	<u>NAME</u>
2	<u>SURNAME</u>
3	<u>GENDER</u>
4	<u>STATE</u>
5	<u>BIRTHDAY</u>
6	<u>HIREDATE</u>
7	<u>DEPT</u>

B3 result

Index	Member
1	[Abigail,Alexander,Alexis, ...]
2	[Adams,Allen,Anderson, ...]
3	[F,M]
4	[Alabama,Arizona,Arkansas, ...]
5	[1968-11-05,1968-11-12,1968-11-24, ...]
6	[1984-04-19,2000-03-07,2000-04-01, ...]
7	[Administration,Finance,HR, ...]

Index	Member
1	1968-11-05
2	1968-11-12
3	1968-11-24
4	1968-11-27
5	1968-12-23

Index	Member
1	1984-04-19
2	2000-03-07
3	2000-04-01
4	2000-05-01
5	2000-06-01

Index	Member
1	<u>Abigail</u>
2	<u>Alexander</u>
3	<u>Alexis</u>
4	<u>Alyssa</u>
5	<u>Amanda</u>

Index	Member
1	<u>Adams</u>
2	<u>Allen</u>
3	<u>Anderson</u>
4	<u>Baker</u>
5	<u>Brown</u>

Index	Member
1	<u>F</u>
2	<u>M</u>

Index	Member
1	<u>Alabama</u>
2	<u>Arizona</u>
3	<u>Arkansas</u>
4	<u>California</u>
5	<u>Colorado</u>

Index	Member
1	<u>Administration</u>
2	<u>Finance</u>
3	<u>HR</u>
4	<u>Marketing</u>
5	<u>Production</u>



Macros are parsed only once, not for loop functions

Calculate the non-repetitive values of non-numeric fields

	A
1	=demo.query("select * from EMPLOYEE")
2	=A1.fname()(A1(1).array().(ifnumber(~)).pos@a(false))
3	=A2.(A1.id(#{A2.~}))
4	=A2.(A1.id(eval(A2.~)))

eval() should be used to generate expressions, which can be parsed many times to get the correct results.

A3, A4 results:

Index	Member
1	[Abigail,Alexander,Alexis, ...]
2	[Abigail,Alexander,Alexis, ...]
3	[Abigail,Alexander,Alexis, ...]
4	[Abigail,Alexander,Alexis, ...]
5	[Abigail,Alexander,Alexis, ...]
6	[Abigail,Alexander,Alexis, ...]
7	[Abigail,Alexander,Alexis, ...]

Index	Member
1	[Abigail,Alexander,Alexis, ...]
2	[Adams,Allen,Anderson, ...]
3	[F,M]
4	[Alabama,Arizona,Arkansa...
5	[1968-11-05,1968-11-12,1...
6	[1984-04-19,2000-03-07,2...
7	[Administration,Finance,H...

The existing data are as follows: a new column is added to generate the calculation results

expression
15X10
5X4
3
2X6X5
22X2X4
3X17

A	
1	=file("E:/txt/expression.txt").import@t()
2	=A1.derive(eval(replace(expression,"X","*")):res)

A1, A2 results:

Index	expression
1	<u>15X10</u>
2	<u>5X4</u>
3	<u>3</u>
4	<u>2X6X5</u>
5	<u>22X2X4</u>
6	<u>3X17</u>

Index	expression	res
1	<u>15X10</u>	150
2	<u>5X4</u>	20
3	<u>3</u>	3
4	<u>2X6X5</u>	60
5	<u>22X2X4</u>	176
6	<u>3X17</u>	51

Employee information is stored in Excel file. Please organize employee information into structured data. Employee information is as follows:

EID	1		
NAME	ZhangYingJ	Gender	Female
Position	Sales Representatives		
Birthday	1968/12/8		
Phone	(010)65559857		
Addr	No. 236 Fuxing Gate, Beijing		
ZipCode	100098		
EID	2		
NAME	WangWei	Gender	Male
Position	Vice President (Sales)		
Birthday	1962/2/19		
Phone	(010)65559482		
Addr	No. 890 Rome Garden, Beijing		
ZipCode	109801		
EID	3		
NAME	LiFang	Gender	Female

	A	B	C
1		=file("E:/txt/employee_info.xlsx").xlsopen()	
2		=create(EID, Name, Gender, Position, Birthday, Phone, Addr, ZipCode)	
3	[B,B,D,B,B,B,B,B]		/Columns for storing employee information in Excel
4	[1,2,2,3,4,5,6,7]		/Rows for storing employee information in Excel
5	for	=A3.(~/A4(#))	/Combine rows and columns in alignment
6		=B5.(eval("A1.xlscell("/~/"/")))	/Resolve strings into expressions
7		if len(B6(1))==0	break
8		>A2.record(B6)	/Put records into table
9		>A4=A4.(~+8)	/next employee

A2 result:

Index	EID	Name	Gender	Position	Birthday	Phone	Addr	ZipCode
1	1	ZhangYingJing	Female	Sales Represe...	1968-12-08	(010)65559857	No. 236 Fuxing...	100098
2	2	WangWei	Male	Vice President ...	1962-02-19	(010)65559482	No. 890 Rome ...	109801
3	3	LiFang	Female	Sales Represe...	1973-08-30	(010)65553421	No. 63 Shaoya...	198033

Example: Select sequences that can form triangles

	A
1	=10.(3.(rand(10)))
2	=A1.select(eval("if(?.(~>0))==?.(~<sum(?\ ~))&&?.(~>0)==?.len().(true))",~))

A1, A2 results:

Index	Member
1	[4,7,5]
2	[7,4,5]
3	[9,3,6]
4	[3,5,5]
5	[2,0,6]
6	[5,8,3]
7	[5,2,9]
8	[1,0,8]
9	[9,0,5]
10	[1,7,3]

Index	Member
1	[4,7,5]
2	[7,4,5]
3	[3,5,5]

Note: Eval () is slow in parsing, so macros are preferred when used.

03

Subprogram

Random generation of two letters as product abbreviation, random generation of 3-6-bit ID as customer abbreviation, create ID, product, customer, sales table

	A	B
1	func	/Generate ID
2		>id=""
3		>A1.run(id=id+char(65+rand(26)))
4		return id
5	func	
6		>B11=B11+1
7		=(rand(1000)+1)*100
8		>A11.insert(0,B11,A5,B5,B7)
9	=5.(func(A1,2))	
10	=10.(func(A1,rand(3)+3))	
11	=create(ID,product,customer,amount)	Fill to the right
12	>100.(func(A5,A9(rand(5)+1),A10(rand(10)+1)))	

A9 result

Index	Member
1	VO
2	MH
3	AA
4	WS
5	YR

A10 result

Index	Member
1	ATQSY
2	DQTQ
3	EUNRM
4	BQD
5	FEQ
6	KOE
7	DGG
8	SWEAV
9	SWBJ
10	OFC

A11 result

Index	ID	product	customer	amount
1	1 VO	BQD		45200
2	2 MH	ATQSY		96100
3	3 YR	DQTQ		69200
4	4 AA	BQD		47700
5	5 MH	SWEAV		27300
6	6 MH	SWEAV		98500
7	7 VO	ATQSY		90200
8	8 WS	OFC		70000
9	9 YR	FEQ		89800
10	10 WS	DQTQ		10400

SPL allows func to refer to external cells internally, such as A11, which is external to func.

Three grid subprograms

createid.dfx

	A
1	/Create ID
2	=""
3	>size.run(A2=A2+char(65+rand(26)))
4	return A2

findnames.dfx

	A
1	Create PName and CName
2	=rand(PNames.len()+1)
3	=rand(CNames.len()+1)
4	return PNames(A2),CNames(A3)

addrecord.dfx

	A
1	/add a record
2	=Table.len()+1
3	=(rand(1000)+1)*100
4	>Table.insert(0,A2,PNAME,CNAME,A3)

Call function calls other grid programs

	A	
1	=5.(call("E:\\esproc_test\\createid.dfx ",2))	/Call createid.dfx to generate product abbreviations
2	=10.(call("E:\\esproc_test\\createid.dfx ",rand(3)+3))	/Call createid.dfx, to generate customer abbreviations
3	=100.(call("E:\\esproc_test\\findnames.dfx",A1,A2))	/Call findnames.dfx to generate the [product, customer] sequence
4	=create(ID,product,customer,amount)	/Create empty table
5	>A3.(call("E:\\esproc_test\\addrecord.dfx",A4,~(1),~(2)))	/Call addre.dfx to insert data to table A4 one by one

A1~A4 results:

Index	Member
1	<u>AA</u>
2	<u>CK</u>
3	<u>EH</u>
4	<u>ML</u>
5	<u>ME</u>

Index	Member
1	<u>SAZV</u>
2	<u>MLE</u>
3	<u>YIRH</u>
4	<u>WAMW</u>
5	<u>RKMTJ</u>
6	<u>XTMT</u>
7	<u>ERI</u>
8	<u>UGQ</u>
9	<u>GFPD</u>
10	<u>ERV</u>

Index	Member
1	<u>[ML,ERI]</u>
2	<u>[CK,WAMW]</u>
3	<u>[ML,WAMW]</u>
4	<u>[ML,XTMT]</u>
5	<u>[EH,UGQ]</u>
6	<u>[AA,SAZV]</u>
7	<u>[EH,UGQ]</u>
8	<u>[AA,RKMTJ]</u>
9	<u>[ML,UGQ]</u>
10	<u>[CK,ERI]</u>

Index	ID	product	customer	amount
1	1	<u>ML</u>	<u>ERI</u>	18000
2	2	<u>CK</u>	<u>WAMW</u>	60200
3	3	<u>ML</u>	<u>WAMW</u>	8200
4	4	<u>ML</u>	<u>XTMT</u>	41100
5	5	<u>EH</u>	<u>UGQ</u>	96600
6	6	<u>AA</u>	<u>SAZV</u>	60700
7	7	<u>EH</u>	<u>UGQ</u>	56600
8	8	<u>AA</u>	<u>RKMTJ</u>	74600
9	9	<u>ML</u>	<u>UGQ</u>	12400
10	10	<u>CK</u>	<u>ERI</u>	82200

In practical application, we often encounter the need to use recursion.
Let's start with a simple example: calculating the factorial of 5.

	A	B
1	func	return if(A1<=0,1,A1*func(A1,A1-1))
2	=func(A1,5)	

A2 result

Value
120

Recursion rule: $n! = n * (n-1)!$
If $n \leq 0$, return 1

Recursive process

Input: $x=5$

Step 1: $\text{func}(5) = 5 * \text{func}(4)$

Step 2: $\text{func}(4) = 4 * \text{func}(3)$

Step 3: $\text{func}(3) = 3 * \text{func}(2)$

Step 4: $\text{func}(2) = 2 * \text{func}(1)$

Step 5: $\text{func}(1) = 1 * \text{func}(0)$

Step 6: $\text{func}(0) = 1$

Result: $5 * 4 * 3 * 2 * 1 = 120$

Euclidean algorithm for solving the maximum common divisor of two numbers

	A	B
1	func	
2		=A1%B1
3		return if(B2==0,B1,func(A1,B1,B2))
4	=func(A1,4557,5115)	

A2 result

Value
93

Recursion rule : The greatest common divisor of a and b = the greatest common divisor of b and b%a
If the remainder is 0, then the divisor is taken

Recursive process

Input: a=4557,b=5115

Step 1: func(4557,5115)=func(5115,5115%4557)

Step 2: func(5115,558)=func(558,5115%558)

Step 3: func(558,93)=93

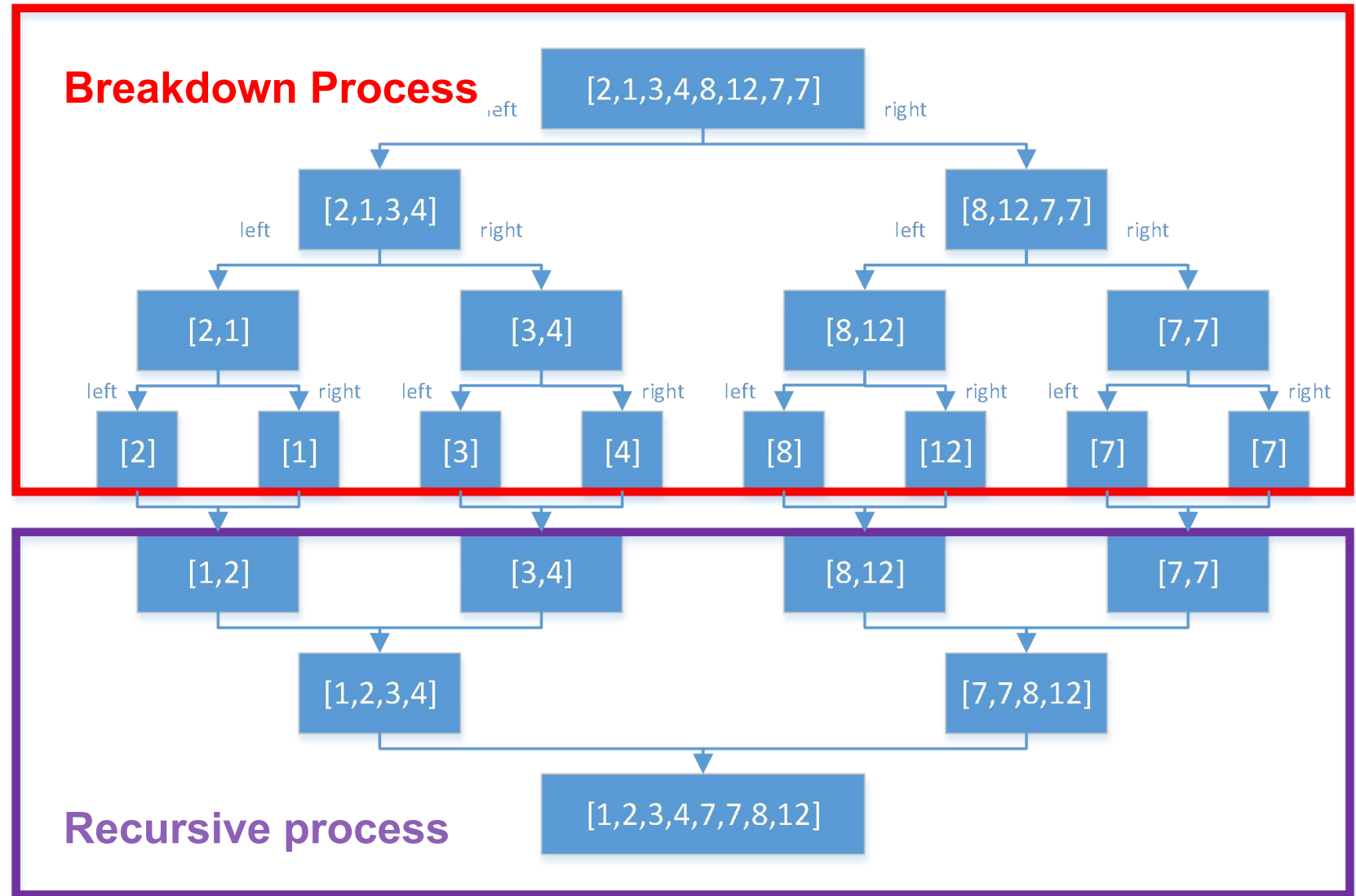
Result: 93

	A	B	C
1	func		
2		=[]	
3		>i=j=1	
4		for (i<=A1.len()&&j<=B1.len())	>if(A1(i)<B1(j),(B2=B2 A1(i),i=i+1),(B2=B2 B1(j),j=j+1))
5		if(i==A1.len(),B2=B2 B1.m(j:))	
6		if(j==B1.len(),B2=B2 A1.m(i:))	
7		return B2	
8	func	if A8.len()<=1	return A8
9		=int(A8.len()\2)	
10		=func(A8,A8.m(:B9))	
11		=func(A8,A8.m(B9+1:))	
12		return func(A1,B10,B11)	
13	=func(A8,[2,1,3,4,8,12,7,7])		

A13 result

Index	Member
1	1
2	2
3	3
4	4
5	7
6	7
7	8
8	12

Recursion rule:
Recursive export : ?



Directories

- ▼ file
 - ▼ FF_2017
 - F_file1
 - FF_2018
 - FF_2019

Files

- FF_file1 1
- FF_file1 2
- FF_file1 3

	A	B
1	=directory@p(path)	/List the full directory of file names under the directory
2	=A1.(file(~).import())	/Import the root directory file
3	=A2.conj()	/Consolidated results
4	=file("d:\\result.txt").export@a(A3)	/Export to file
5	=directory@dp(path)	/List the directories under the directory
6	>A5.(call("E:/esproc_test/readfiles.dfx",~))	/Call the script recursively

- FF_file1 1
- FF_file1 2
- FF_file1 3
- FF_file2 4
- FF_file2 5
- FF_file2 6
- FF_file3 7
- FF_file3 8
- FF_file3 9
- FF_file1/F_file1 1
- FF_file1/F_file1 2
- FF_file1/F_file1 3
- FF_file1/F_file2 4
- FF_file1/F_file2 5
- FF_file1/F_file2 6
- FF_file1/F_file3 7
- FF_file1/F_file3 8
- FF_file1/F_file3 9
- FF_file1/F_file1/file1 1

A1

Index	Member
1	D:\file\FF_file1.txt
2	D:\file\FF_file2.txt
3	D:\file\FF_file3.txt

A2

Index	Member
1	[[FF_file1,1],[FF_file1,2],[FF_file1,3]]
2	[[FF_file2,4],[FF_file2,5],[FF_file2,6]]
3	[[FF_file3,7],[FF_file3,8],[FF_file3,9]]

A3

Index	_1	_2
1	FF_file1	1
2	FF_file1	2
3	FF_file1	3
4	FF_file2	4
5	FF_file2	5
6	FF_file2	6
7	FF_file3	7
8	FF_file3	8
9	FF_file3	9

A5

Index	Member
1	D:\file\FF_2017
2	D:\file\FF_2018
3	D:\file\FF_2019

04

Parallel

Multi-threading to identify the oldest employees in Sales, R&D, Finance, and Production

	A	B
1	<code>\$(demo) select * from EMPLOYEE</code>	<code>[Sales,R&D,Finance,Production]</code>
2	<code>fork B1</code>	<code>=A1.select(DEPT==A2)</code>
3		<code>=B2.minp(BIRTHDAY)</code>
4		<code>return B3</code>

A2 result

Index	EID	NAME	SURNAME	GENDER	STATE	BIRTHDAY	HIREDATE	DEPT	SALARY
1	331	Tyler	Williams	M	Florida	1968-12-23	2001-09-01	Sales	6500
2	191	David	Johnson	M	Florida	1969-01-31	2007-01-01	R&D	5000
3	32	Andrew	Williams	M	Texas	1971-08-27	2001-08-27	Finance	11000
4	296	Olivia	Johnson	F	Ohio	1968-11-05	2006-11-01	Production	10000

Count the number of all words in Harry Potter Novels

Parallel

	A	B
1	=directory@p("D:/test/Harry Potter 1-7")	=now()
2	for A1	=file(A2).read().words()
3		=@[B2.groups(lower(~):Word;count(~):Count)]
4	=B3.merge(Word)	
5	=A4.groups@o(Word;sum(Count):Count).sort@z(Count)	
6	=interval@ms(B1,now())	

Novel files

- 1.Harry Potter and the Sorcerer's Stone.txt
- 2.Harry Potter and The Chamber Of Secrets.txt
- 3.Harry Potter and the Prisoner of Azkaban.txt
- 4.Harry Potter and the Goblet of Fire.txt
- 5.Harry Potter and the Order of the Phoenix.txt
- 6.Harry Potter and The Half-Blood Prince.txt
- 7.《哈利波特与死圣》英文版HP7_[csbeyond].txt

A5, A6 results

Index	Word	Count
1	<u>the</u>	51889
2	<u>and</u>	27518
3	<u>to</u>	26996
4	<u>he</u>	22203
5	<u>of</u>	21847
6	<u>a</u>	21043
7	<u>harry</u>	18158
8	<u>was</u>	15637
9	<u>you</u>	14627
10	<u>it</u>	14489

Value
981

fork Multi-threads

	A	B
1	=directory@p("D:/test/Harry Potter 1-7")	=now()
2	fork A1	=file(A2).read().words()
3		=B2.groups(lower(~):Word;count(~):Count)
4	=A2.merge(Word)	
5	=A4.groups@o(Word;sum(Count):Count).sort@z(Count)	
6	=interval@ms(B1,now())	

A5 result

Index	Word	Count
1	<u>the</u>	51889
2	<u>and</u>	27518
3	<u>to</u>	26996
4	<u>he</u>	22203
5	<u>of</u>	21847
6	<u>a</u>	21043
7	<u>harry</u>	18158
8	<u>was</u>	15637
9	<u>you</u>	14627
10	<u>it</u>	14489

Value
607

Take out Sales, Marketing, and Productions employees from the database

	A	B
1	[Sales,Marketing,Production]	
2	=now()	
3	for A1	=connect("demo")
4		=@ B3.query@x("select * from EMPLOYEE where DEPT=?",A3)
5	=interval@ms(A2,now())	

B4, A5 results:

Index	EID	NAME	SURNAME	GENDER	STATE	BIRTHDAY	HIREDATE	DEPT	SALARY
1	3	Rachel	Johnson	F	New Mexico	1970-12-17	2010-12-01	Sales	9000
2	6	Matthew	Johnson	M	California	1984-07-07	2005-07-07	Sales	11000
3	7	Alexis	Smith	F	Illinois	1972-08-16	2002-08-16	Sales	9000
4	11	Jacob	Moore	M	Texas	1974-12-16	2004-12-16	Sales	12000
5	12	Jessica	Davis	F	New York	1980-09-11	2008-09-11	Sales	7000

Value
346

Take out Sales, Marketing, and Productions employees from the database

	A	B
1	[Sales,Marketing,Production]	
2	=now()	
3	fork A1	=connect("demo")
4		=B3.query@x("select * from EMPLOYEE where DEPT=?",A3)
5	=A3.conj()	
6	=interval@ms(A2,now())	

A5, A6 results:

Index	EID	NAME	SURNAME	GENDER	STATE	BIRTHDAY	HIREDATE	DEPT	SALARY
1	3	Rachel	Johnson	F	New Mexico	1970-12-17	2010-12-01	Sales	9000
2	6	Matthew	Johnson	M	California	1984-07-07	2005-07-07	Sales	11000
3	7	Alexis	Smith	F	Illinois	1972-08-16	2002-08-16	Sales	9000
4	11	Jacob	Moore	M	Texas	1974-12-16	2004-12-16	Sales	12000
5	12	Jessica	Davis	F	New York	1980-09-11	2008-09-11	Sales	7000

Value
132



THANKS