Subject: stable sort bug.. or looks like it
Posted by aftershock on Sun, 21 May 2017 20:30:56 GMT
View Forum Message <> Reply to Message

Have a look
VectorMap<String, int> involved_nodes;

```
 DUMPM ( involved_nodes );
 StableSort ( involved_nodes.Begin(), involved_nodes.End(), lip );

 DUMPM ( involved_nodes );
```

Stable sort does not seem copy key value pairs correctly for maps.

before
involved_nodes:
[0] = (tester_withdraw_when) 0
[1] = (tester_withdraw_amount) 2
[2] = (tester_balance) 3
[3] = (trading_buy_open_start_hour) 12
[4] = (trading_buy_open_end_hour) 13
[5] = (trading_sell_open_start_hour) 18
[6] = (trading_sell_open_end_hour) 19
[7] = (buy_closewhenprofit) 38
[8] = (buy_stepsize) 43
[9] = (buy_stopdistance) 47
[10] = (sell_stepsize) 51
[11] = (sell_closewhenprofit) 52
[12] = (buy_close_after_minutes) 180
[13] = (sell_close_after_minutes) 181
[14] = (sell_stopdistance) 216
[15] = (buy_stop_fallback) 221
[16] = (sell_stop_fallback) 225
[17] = (buy_rsi_timeframe) 229
[18] = (sell_rsi_timeframe) 230
[19] = (buy_rsi_upper_limit) 367
[20] = (sell_rsi_lower_limit) 368
after incorrect result
involved_nodes:
[0] = (tester_withdraw_when) 3
[1] = (tester_withdraw_amount) 0
[2] = (tester_balance) 2
[3] = (trading_buy_open_start_hour) 19
[4] = (trading_buy_open_end_hour) 13
[5] = (trading_sell_open_start_hour) 18
[6] = (trading_sell_open_end_hour) 12
[7] = (buy_closewhenprofit) 38

[8] = (buy_stepsize) 47
[9] = (buy_stopdistance) 51
[10] = (sell_stepsize) 216
[11] = (sell_closewhenprofit) 221
[12] = (buy_close_after_minutes) 229
[13] = (sell_close_after_minutes) 43
[14] = (sell_stopdistance) 52
[15] = (buy_stop_fallback) 180
[16] = (sell_stop_fallback) 181
[17] = (buy_rsi_timeframe) 225
[18] = (sell_rsi_timeframe) 230
[19] = (buy_rsi_upper_limit) 367
[20] = (sell_rsi_lower_limit) 368

This is what it should be.....
involved_nodes:
[0] = (tester_withdraw_when) 0
[1] = (tester_withdraw_amount) 2
[2] = (tester_balance) 3
[3] = (trading_buy_open_start_hour) 12
[4] = (trading_buy_open_end_hour) 13
[5] = (trading_sell_open_start_hour) 18
[6] = (trading_sell_open_end_hour) 19
[7] = (buy_closewhenprofit) 38
[8] = (buy_stepsize) 43
[9] = (buy_stopdistance) 47
[10] = (sell_stepsize) 51
[11] = (sell_closewhenprofit) 52
[12] = (buy_close_after_minutes) 180
[13] = (sell_close_after_minutes) 181
[14] = (sell_stopdistance) 216
[15] = (buy_stop_fallback) 221
[16] = (sell_stop_fallback) 225
[17] = (buy_rsi_timeframe) 229
[18] = (sell_rsi_timeframe) 230
[19] = (buy_rsi_upper_limit) 367
[20] = (sell_rsi_lower_limit) 368

Thanks.. Can you fix this?

A.

Subject: Re: stable sort bug.. or looks like it
Posted by mirek on Mon, 22 May 2017 10:00:42 GMT

View Forum Message <> Reply to Message

aftershock wrote on Sun, 21 May 2017 22:30Have a look
VectorMap<String, int> involved_nodes;

```
 DUMPM ( involved_nodes );
 StableSort ( involved_nodes.Begin(), involved_nodes.End(), lip );

 DUMPM ( involved_nodes );
```

Stable sort does not seem copy key value pairs correctly for maps.

All Sorts, when used like this, only sort values, as begin/end here return iterators to values only (the value subvector).

I think that to do what you need, you can use [Stable]SortByValues and then StableSortByKeys.

(Perhaps we might add [Stable]SortBy[KeysAndValues|ValuesAndKeys] in future...)

Mirek

Subject: Re: stable sort bug.. or looks like it
Posted by aftershock on Mon, 22 May 2017 18:45:28 GMT
View Forum Message <> Reply to Message

stablesortbykey is better... but then value information is lost..
Only a new function would help.
in this case [Stable]SortBy[KeysAndValues| could help as I have custom comparison function..
The aim is to keep pair/value relationship..

Subject: Re: stable sort bug.. or looks like it
Posted by mirek on Mon, 22 May 2017 23:24:48 GMT
View Forum Message <> Reply to Message

aftershock wrote on Mon, 22 May 2017 20:45stablesortbykey is better... but then value information is lost..

What do you mean by "lost"? Not sorted by, or damaged by SortByKeys? (that would be a bug).

Subject: Re: stable sort bug.. or looks like it
Posted by aftershock on Tue, 23 May 2017 09:18:18 GMT
View Forum Message <> Reply to Message

I mean
before the operation stablesortbykey

we have key/value pair
(tester_withdraw_when) 0

after the operation

(tester_withdraw_when) 3

So the unity of key/pair is not kept... the key loses its value to something else.

Subject: Re: stable sort bug.. or looks like it
Posted by mirek on Tue, 23 May 2017 12:19:21 GMT
View Forum Message <> Reply to Message

aftershock wrote on Tue, 23 May 2017 11:18I mean
before the operation stablesortbykey

we have key/value pair
(tester_withdraw_when) 0

after the operation

(tester_withdraw_when) 3

So the unity of key/pair is not kept... the key loses its value to something else.

That is really weird. I have just tested with this code:

```
#include <Core/Core.h>

using namespace Upp;

CONSOLE_APP_MAIN
{
 VectorMap<int, String> map;

 for(int i = 0; i < 100; i++) {
  int n = Random(200);
  map.Add(n, FormatIntRoman(n) + " " + AsString(n) + " " + AsString(i));
 }
```

```
 DUMPM(map);

 StableSortByKey(map);

 DUMPM(map);
}
```

and everything seems 100% fine.

Can you post your testcase please?

Mirek

Subject: Re: stable sort bug.. or looks like it
Posted by aftershock on Wed, 24 May 2017 09:46:49 GMT
View Forum Message <> Reply to Message

I reexamined the data.

You are right , it works.. there is nothing to do.
I must have misread the data.
