Subject: SortedIndex and Less
Posted by keltor on Tue, 10 Sep 2013 15:34:38 GMT
View Forum Message <> Reply to Message

Hey there,

I am finally stepping into NTL and so far I like it a lot, much nicer than STL, I think. I have encountered a hurdle though.

I would like to sort a class. I provide what I thought was enough data to appease the NTL gods, but it seems I need some more stuff.

Here's a sample of my simple test program:

```
struct rho : Moveable<rho> {
  int x,y,z;

  rho(int x,int y,int z) : x(x), y(y), z(z) {}
  rho() {}
 }r;

 unsigned GetHashValue(const rho& p)
 {
  return CombineHash(p.x, p.x, p.z);
 }

 bool operator==(const rho& a, const rho& b)
 {
  return a.x == b.x && a.y == b.y && a.z == b.z;
 }

 bool operator < (const rho& a, const rho& b){
  return a.x == b.x ? (a.y == b.y ? a.z < b.z : a.y < b.y) : a.x < b.x;
 }
```

Index<rho> works fine (without the operator < part), but SortedIndex<rho> does not. I looked at the StdLess<T> code and it seems that all it does is to provide a simple return a < b operation. What am I missing?

Thanks,

Kel

Subject: Re: SortedIndex and Less
Posted by mirek on Sat, 21 Sep 2013 17:12:24 GMT
View Forum Message <> Reply to Message

Not sure what went wrong, I have tried this:

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

using namespace Upp;

struct rho : Moveable<rho> {
 int x,y,z;

 rho(int x,int y,int z) : x(x), y(y), z(z) {}
 rho() {}

 String ToString() const { return AsString(x) + ' ' + AsString(y) + ' ' + AsString(z); }
};

bool operator < (const rho& a, const rho& b){
 return a.x == b.x ? (a.y == b.y ? a.z < b.z : a.y < b.y) : a.x < b.x;
}

CONSOLE_APP_MAIN{
 StdLogSetup(LOG_FILE);

 SortedIndex<rho> data;
 data.Add(rho(1, 2, 3));
 data.Add(rho(1, 1, 1));
 data.Add(rho(1, 2, 0));

 DDUMPC(data);
}
```

and it seems to work fine...

Note: you can use CombineCompare helper:

```
bool operator < (const rho& a, const rho& b){
 return CombineCompare(a.x, b.x)(a.y, b.y)(a.z, b.z) < 0;
}
```

Mirek

Subject: Re: SortedIndex and Less
Posted by keltor on Tue, 24 Sep 2013 14:08:03 GMT

View Forum Message <> Reply to Message

I have not updated my U++ in a while, that's probably the culprit if it works for you.

Thanks for the help and the tip, Mirek.

Subject: Re: SortedIndex and Less
Posted by keltor on Tue, 24 Sep 2013 15:15:53 GMT
View Forum Message <> Reply to Message

Thanks to the example above, I have found why I was getting an error. I pretty much wrote the same code, but I also had the following line in my test:

data.FindAdd(rho(1, 2, 3));

which gives an error. But the compiler points at the definition of data instead of at that line, so I thought that my object had something missing.

However, this opens a new question: how to fix this? It looks as if both FindAdd and Add have the same paradigm, yet I get

```
c:\upp\uppsrc\core\InVector.hpp(237) : error C2664: 'rho &Upp::Vector<T>::Add(const T &)' :
cannot convert parameter 1 from 'int' to 'const
  rho &'
        with
        [
            T=rho
        ]
        Reason: cannot convert from 'int' to 'const rho'
```

Is this a bug, or am I doing something stupid here?

Edit: Incidentally, I am now using build 6254. Not the latest, I know, but not very old either.

Subject: Re: SortedIndex and Less
Posted by mirek on Tue, 24 Sep 2013 15:28:53 GMT
View Forum Message <> Reply to Message

Seriously, posting more complete code snippet would really help us to help you

Subject: Re: SortedIndex and Less
Posted by keltor on Wed, 25 Sep 2013 06:12:59 GMT
View Forum Message <> Reply to Message

OK, sorry. But really all I did was to add that line to the bottom of your code. Here's the full program:

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

using namespace Upp;

struct rho : Moveable<rho> {
 int x,y,z;

 rho(int x,int y,int z) : x(x), y(y), z(z) {}
 rho() {}

 String ToString() const { return AsString(x) + ' ' + AsString(y) + ' ' + AsString(z); }
};

bool operator < (const rho& a, const rho& b){
 return a.x == b.x ? (a.y == b.y ? a.z < b.z : a.y < b.y) : a.x < b.x;
}

CONSOLE_APP_MAIN{
 StdLogSetup(LOG_FILE);

 SortedIndex<rho> data;
 data.Add(rho(1, 2, 3));
 data.Add(rho(1, 1, 1));
 data.Add(rho(1, 2, 0));
 data.FindAdd(rho(1, 2, 3));

 DDUMPC(data);
}
```

Thanks for the help Mirek

Subject: Re: SortedIndex and Less
Posted by mirek on Wed, 25 Sep 2013 07:07:16 GMT
View Forum Message <> Reply to Message

Well, actually, this was a bug in U++. I am sorry about that, In* containers are quite new and it appears that nobody tested them with "user" types yet (the bug does not affect it with String or numberic types).

Now fixed.

Mirek

Subject: Re: SortedIndex and Less
Posted by keltor on Wed, 25 Sep 2013 08:18:22 GMT
View Forum Message <> Reply to Message

Great! Thanks for all your effort and I'm also glad that my seemingly simple question helped to find a bug.