Subject: Algorithms and Vectors
Posted by 281264 on Tue, 17 Aug 2010 20:58:42 GMT
View Forum Message <> Reply to Message

Hi,

I have a query regarding the usage of Vectors and U++ algorithms. The case is as follows: I am using Vector to manage a list of objects; the structure of the objects in question is rather simple: 3 double variables, one unsigned integer and a bool variable. Is it possible to use Find or FindBinary (or any other algorithm) to make a search within the Vector of structs?

Cheers,

Javier

Subject: Re: Algorithms and Vectors
Posted by mdelfede on Tue, 17 Aug 2010 21:45:24 GMT
View Forum Message <> Reply to Message

Yep, you should use a predicate :

```
class Predicate
{
    public:
        bool operator() (const M& lhs, const M& rhs) {
            return lhs.somedata < rhs.somedata;
}
```

then

```
M myVar;  // M is your data class/struct
Vector<M> myVect;
FindBinary(myVect, myVar, Predicate())
```

Ciao

Max

Subject: Re: Algorithms and Vectors
Posted by 281264 on Wed, 18 Aug 2010 10:30:30 GMT

Thanks. This is an easy example, but it does not work

```
struct A:Moveable<A>{

 int a;
 A(int x){a=x;}
};

class Predicate
{

public:

 bool operator()(const A& lhs, const A& rhs){
  return lhs.a==rhs.a;
 }

};


int main(){

 A vector_b(1);
 Vector<A> vector_a;

 vector_a.Add(1);
 vector_a.Add(2);

 FindBinary(vector_a,vector_b,Predicate());

 return 0;
}
```

why?

Cheers,

Javier

Subject: Re: Algorithms and Vectors
Posted by dolik.rce on Wed, 18 Aug 2010 14:06:44 GMT

Hi Javier,

The predicate is basicaly a less than function, not equality. So the code should beclass Predicate{
public:
 bool operator()(const A& lhs, const A& rhs){
  return lhs.a<rhs.a;
 }
};

Honza

Subject: Re: Algorithms and Vectors
Posted by 281264 on Wed, 18 Aug 2010 15:20:43 GMT
View Forum Message <> Reply to Message

Hi Honza,

Roger ! however I have noticed the following:

1.- even with < in the class the code does not compile, why?
2.- if I replace the class by the function:bool operator<(const A& lhs, const A& rhs){
  return lhs.a<rhs.a;
}
then, it does compile but it does not behave as expected (it returns -1).So it does not work.

instance a==5;

Note: I have tested Sort algorithm and works fine.

Any suggestions?

Thanks,

Javier

Subject: Re: Algorithms and Vectors
Posted by dolik.rce on Wed, 18 Aug 2010 16:52:29 GMT
View Forum Message <> Reply to Message

Oh, ok, I see... Well, I don't have right now time to test compiling the code, but I can tell you right away, that you actually don't need this. Find(), FindBinary(), FindUpperBound() etc are doing something little else. They return index where the element should be inserted to keep the array sorted. Check the manual for details.

The simple way to find something, is to just go through the array in for cycle and look for the element you need. For simple struct like yours this should be pretty fast, especially if you check only value of one member.

If you need somewhat complex checking or exact equality of members, I would recommend you using ArrayIndex instead of Array. The only requirement is that you have to implement GetHashValue() function. If I remember correctly that is described quite good in Core Values Tutorial. Then you can search just using myindex.Find(myobject).

Sorry for not giving deeper explanation, I'm in hurry now

Honza

Subject: Re: Algorithms and Vectors
Posted by 281264 on Wed, 18 Aug 2010 17:49:24 GMT
View Forum Message <> Reply to Message

Your explanation is fine. I agree with you in relation with the usage of a for-loop; it is the simplest

Cheers!

Javier

Subject: Re: Algorithms and Vectors
Posted by mr_ped on Fri, 20 Aug 2010 19:05:56 GMT
View Forum Message <> Reply to Message

The FindBinary actually should return index of the searched value, in your example "0", or -1 if the value is not found.

So something's wrong if it does not work.

But the vector must be already sorted by that less predicate before calling it (as far as I can tell, Vector {1,2} is sorted, so that piece of code still looks fine).

Maybe I should start TheIDE a try it myself... (reluctant to do it and compile, I'm on very slow PC)

Subject: Re: Algorithms and Vectors
Posted by 281264 on Mon, 23 Aug 2010 20:33:11 GMT
View Forum Message <> Reply to Message

To me, it is a bit weird why the code does not work.

Any volunteer to shed some light?

Cheers,

Javier
