search

Search

[c++] What is the point of make_heap? [stl]

Can someone please tell me the point of the *stl* heap functions like make_heap? Why would anyone ever use them? Is there a practical use?

c++ stl language-design

Instagram Feed

7 Recent answers

1. If you want to make a priority queue out from a list, well, you can use make_heap:
Internally, a heap is a tree where each node links to values not greater than its own value. In heaps generated by make_heap, the specific position of

an element in the tree rather than being determined by memory-consuming links is determined by its absolute position in the sequence, with *first being always the highest value in the heap.
Heaps allow to add or remove elements from it in logarithmic time by using functions push_heap and pop_heap, which preserve its heap properties.

2. Your direct question would be well-answered by a class in algorithms and data structures. Heaps are used all over the place in algorithms in computer science. To quote from the make_heap function linked below, "a heap is a tree where each node links to values not greater than its own value." While there are lots of applications for a heap, the one that I use most frequently is in search problems when you want to keep track of a sorted list of N values efficiently.
I had similar confusion to yours when I first encountered the *stl* heap functions. My question was a little bit different though. I wondered "Why isn't the *stl* heap in the same class of data structures as std::vector?" I thought that it should work like this:

```
std::heap< int > my_heap;
my_heap.heap_insert( 7 );
my_heap.heap_insert( 3 );
```

The idea behind the *stl* heap functions though is that they allow you to make a heap data structure out of several different underlying *stl* containers, including std::vector. This can be really useful if you want to pass around the container for use elsewhere in your programs. It's also a little bit nice, because you can choose the underlying container of your heap if you so choose to use a something other than std::vector. All you really need are the following:

```
template <class RandomAccessIterator>
void make_heap ( RandomAccessIterator first, RandomAccessIterator
last );
```

This means that you can make lots of different containers into a heap A comparator is also optional in the method signature, you can read more about the different things that you can try in the *stl* pages for the make_heap function.
Links:
Heap Data Structure make_heap function

3. In addition to the above, the *stl*'s sorting algorithm is introsort, which is a mixture of quicksort and heapsort (it fails over from quicksort to heapsort if the former is doing poorly). make_heap creates a heap structure, which is needed for running heapsort, which is needed for introsort.

4. You are supposed to use `std::make_heap()` along with `std::push_heap()` and `std::pop_heap()` to maintain a binary heap on top of a vector or array; the latter two functions maintain the heap invariant. You can also use `std::heap_sort()` to sort such a heap. While it is true that you could use `std::priority_queue` for a priority queue, it doesn't let you get at the insides of it, which perhaps you want to do. Also, `std::make_heap()` and `std::heap_sort()` together make a very simple way of doing heapsort in **c++**.

5. There are essentially two ways to construct a [binary] heap: create an empty heap and insert each element into it one at a time, or take a range of values and heapify them.
Each push operation on a heap takes O(logn) time so if you are pushing N items onto a heap it will take O(NlogN) time. However to build a binary heap from an array of values takes only O(N) time.
Thus it makes more sense to insert each element into an array (or other container that supports random access iterators) and then call make_heap() on the array than it does to maintain the heap structure while inserting.

6. If you want to make a priority queue out from a list, well, you can use make_heap:
Internally, a heap is a tree where each node links to values not greater than its own value. In heaps generated by make_heap, the specific position of an element in the tree rather than being determined by memory-consuming links is determined by its absolute position in the sequence, with *first being always the highest value in the heap.
Heaps allow to add or remove elements from it in logarithmic time by using functions push_heap and pop_heap, which preserve its heap properties.

7. `std::make_heap` should almost never be used in practice. While it is true that heaps are useful for priority queues, that doesn't explain why you would want to manually maintain the structure. `std::priority_queue` has a much more useful interface if all you need is a priority queue.
If you use `make_heap` and its siblings directly, you have to make sure to use them every single time you make a change to the underlying container. I have seen them used two or three times and every single time they were used incorrectly.
I have only used the heap operations directly once myself, because I needed to use a vector as a priority queue for a while and then sort it. You will most likely never need `std::make_heap`.
If you need a priority queue with the ability to change elements, you can

use `std::set`. You can get the smallest or largest element with `*s.begin()` or `*s.rbegin()` respectively and update an element by removing the old value and inserting the new one.

8. In addition to the above, the *stl*'s sorting algorithm is introsort, which is a mixture of quicksort and heapsort (it fails over from quicksort to heapsort if the former is doing poorly). make_heap creates a heap structure, which is needed for running heapsort, which is needed for introsort.

9. They are used to construct and maintain the Heap data structure

10. There are essentially two ways to construct a [binary] heap: create an empty heap and insert each element into it one at a time, or take a range of values and heapify them.
Each push operation on a heap takes O(logn) time so if you are pushing N items onto a heap it will take O(NlogN) time. However to build a binary heap from an array of values takes only O(N) time.
Thus it makes more sense to insert each element into an array (or other container that supports random access iterators) and then call make_heap() on the array than it does to maintain the heap structure while inserting.

About Tag

c++

C++ is a general-purpose programming language based on C. Use this tag for questions about code compiled with a C++ compiler.

stl

The Standard Template Library, or STL, is a C++ library of generic containers, iterators, algorithms, and function objects. When C++ was standardised, large parts of the STL were adopted into the Standard Library, and these parts in the Standard Library are also sometimes referred to collectively ...

language-design

A tag for questions related to the design of any aspect of programming languages.

Related question

3
Votes
5
Comments

Mysql strip time component from datetime

I need to do a date comparison in Mysql without taking into account the time component i.e. i need to convert '2008-11-05 14:30:00' to '2008-11-05' Currently i am doing this: SELECT from_days(to_days(my_date)) Is there a proper way of doing this?...

3
Votes
3
Comments

Set a database value to null with a SqlCommand + parameters

I was previously taught today how to set parameters in a SQL query in .NET in this answer (click). Using parameters with values are fine, but when I try to set a field in the database to null I'm unsuccessful. Either the method thinks I am not setting a valid parameter or not specifying a parameter. e.g. Dim dc As New SqlCommand("UPDATE Activities SET [Limit] = @Limit WHERE [Activity] = @Activity",...

3
Votes
2
Comments

Double curved shape

I am currently attempting to generate a 'wavy ghostly bottom' shape. This shape contains two double curves: Although the bottom part of this image I think portrays it in better imagery. My Code My Current Attempt to generate this shape was using pseudo elements and overflow: hidden, although this does not allow for a gradient background (would require a plain background): Attempt 1 - Using...

17
Votes
5
Comments

Read a file/URL line-by-line in Swift

I am trying to read a file given in an NSURL and load it into an array, with items separated by a newline character . Here is the way I've done it so far: var possList: NSString? = NSString.stringWithContentsOfURL(filePath.URL) as? NSString if var list = possList { list = list.componentsSeparatedByString(" ") as NSString[] return list } else { //return empty list } I'm not very...

4
Votes
1
Comments

What is the purpose of XCTestCase's setUp method?

Per the comment within the default template for XCTestCase regarding setUp : Put setup code here; it will be run once, before the first test case. However, in XCTestCase.h, the comment above setUp states differently: Setup method called before the invocation of each test method in the class. To confirm the actual behavior, I put an NSLog withinsetUp to count how many times it was called: static...

7
Votes
3
Comments

Thoughts on using Zurb's Foundation+AngularJS?

I know this sounds like a pretty generic question, but I have already made a research and there is not much information out there, other than these two seem not to "play" very well together. I am referring to the latest version of Foundation, 5, which just got released. If AngularJS can work with jQuery, and Foundation is using jQuery, what is it that causes the problem, if there is a problem at...

9
Votes
4
Comments

At underscore js, Can I get multiple columns with pluck method after input where method as linq select projection

var people = [{firstName : "Thein", city : "ny", qty : 5}, {firstName : "Michael", city : "ny", qty : 3}, {firstName : "Bloom", city : "nj", qty : 10}]; var results=_.pluck(_.where(people, {city : "ny"}), 'firstName'); For example : I need firstName and qty....

22
Votes
1
Comments

Attack Protection for iOS In-App Purchases

Apple's iOS in-app purchase system has been attacked in the past by people who have tricked apps into giving them content for free. They have since improved the systems involved to try to limit this kind of thing. I've read through the StoreKit reference documents available from Apple and I have a general idea of the workflow and the checks that need to be done, and so on. However, there may be...

5
Votes
4
Comments

Post increment operator not incrementing in for loop

I'm doing some research about Java and find this very confusing: for (int i = 0; i < 10; i = i++) { System.err.print("hoo... "); } This is never ending loop! Anybody has good explanation why such thing happens?...

6
Votes
5
Comments

UIBarButtonItem: How can I find its frame?

I have a button in a toolbar. How can I grab its frame? Do UIBarButtonItems not have a frame property?...

Your Answer

Tell us as much as you can without sharing any personal information. Please try not to use any insulting vocabulary - even if it's hard. Otherwise we are obliged to remove your comment.

Add a answer here

Anti-spam verification:

Post Your Answer