heaps

0.3.0

Generated by Doxygen 1.8.17

Chapter 1

Class Index

1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

Chapter 3

Class Documentation

3.1 array< T > Struct Template Reference

Classes

- struct BinaryHeap

Public Member Functions

- array (int len)
- T & operator[ ] (int i)
- array< T > & operator= (array< T > &b)
- int size ()
- T get (int i)
- T set (int i, T x)
- void add (int i, T x)
- T remove (int i)
- void resize ()

Public Attributes

- T * a
- int length
- int n

3.1.1 Detailed Description

template<typename T>
struct array< T >

Definition at line 17 of file main.cpp.

3.1.2 Constructor & Destructor Documentation

3.1.2.1 array()

```
template<typename T >
array< T >::array (
            int len )  [inline]
```

Definition at line 21 of file main.cpp.

```
21                  {
22 length = len;
23 a = new T[length];
24 }
```

3.1.3 Member Function Documentation

3.1.3.1 add()

```
template<typename T >
void array< T >::add (
            int i,
            T x )  [inline]
```

Definition at line 61 of file main.cpp.

```
61                          {
62 if (n + 1 > a.length) resize();
63 for (int j = n; j > i; j--)
64 a[j] = a[j - 1];
65 a[i] = x;
66 n++;
67 }
```

3.1.3.2 get()

```
template<typename T >
T array< T >::get (
            int i )  [inline]
```

Definition at line 52 of file main.cpp.

```
52                  {
53 return a[i];
54 }
```

3.1.3.3 operator=()

```
template<typename T >
array<T>& array< T >::operator= (
            array< T > & b )  [inline]
```

Definition at line 31 of file main.cpp.

```
31                                  {
32 if (a != NULL) delete[] a;
33 a = b.a;
34 b.a = NULL;
35 length = b.length;
36 return *this;
37 }
```

3.1.3.4 operator[]()

```
template<typename T >
T& array< T >::operator[] (
            int i )  [inline]
```

Definition at line 26 of file main.cpp.

```
26                          {
27 assert(i >= 0 && i < length);
28 return a[i];
29 }
```

3.1.3.5 remove()

```
template<typename T >
T array< T >::remove (
            int i )  [inline]
```

Definition at line 69 of file main.cpp.

```
69                    {
70 T x = a[i];
71 for (int j = i; j < n - 1; j++)
72 a[j] = a[j + 1];
73 n--;
74 if (a.length >= 3 * n) resize();
75 return x;
76 }
```

3.1.3.6 resize()

```
template<typename T >
void array< T >::resize ( )  [inline]
```

Definition at line 78 of file main.cpp.

```
78                {
79 array<T> b(max(2 * n, 1));
80 for (int i = 0; i < n; i++)
81 b[i] = a[i];
82 a = b;
83 }
```

3.1.3.7 set()

```
template<typename T >
T array< T >::set (
            int i,
            T x )  [inline]
```

Definition at line 55 of file main.cpp.

```
55                  {
56 T y = a[i];
57 a[i] = x;
58 return y;
59 }
```

3.1.3.8 size()

```
template<typename T >
int array< T >::size ( )  [inline]
```

Definition at line 48 of file main.cpp.

```
48           {
49 return n;
50 }
```

3.1.4 Member Data Documentation

3.1.4.1 a

```
template<typename T >
T* array< T >::a
```

Definition at line 18 of file main.cpp.

3.1.4.2 length

```
template<typename T >
int array< T >::length
```

Definition at line 19 of file main.cpp.

3.1.4.3 n

```
template<typename T >
int array< T >::n
```

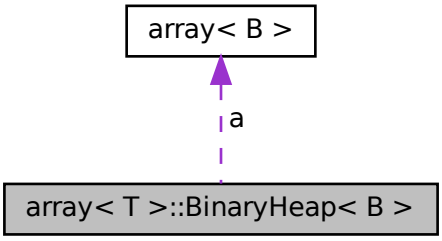
Definition at line 47 of file main.cpp.

The documentation for this struct was generated from the following file:

- /home/benson/CPTR227/Heaps/src/main.cpp

3.2 array< T >::BinaryHeap< B > Struct Template Reference

Collaboration diagram for array< T >::BinaryHeap< B >:



Public Member Functions

- int left (int i)
- int right (int i)
- int parent (int i)
- bool add (B x)
- void bubbleUp (int i)
- B remove ()
- void trickleDown (int i)

Public Attributes

- array< B > a
- int n

3.2.1 Detailed Description

template<typename T>
template<typename B>
struct array< T >::BinaryHeap< B >

Definition at line 97 of file main.cpp.

3.2.2 Member Function Documentation

3.2.2.1 add()

```
template<typename T >
template<typename B >
bool array< T >::BinaryHeap< B >::add (
            B x )  [inline]
```

Definition at line 112 of file main.cpp.

```
112                     {
113             if (n + 1 > a.length) resize();
114             a[n++] = x;
115             bubbleUp(n-1);
116             return true;
117         }
```

3.2.2.2 bubbleUp()

```
template<typename T >
template<typename B >
void array< T >::BinaryHeap< B >::bubbleUp (
            int i )  [inline]
```

Definition at line 119 of file main.cpp.

```
119                             {
120         int p = parent(i);
121         while (i > 0 && compare(a[i], a[p]) < 0) {
122             a.swap(i,p);
123             i = p;
124             p = parent(i);
125             }
126         }
```

3.2.2.3 left()

```
template<typename T >
template<typename B >
int array< T >::BinaryHeap< B >::left (
            int i )  [inline]
```

Definition at line 99 of file main.cpp.

```
99                      {
100             return 2*i + 1;
101         }
```

3.2.2.4 parent()

```
template<typename T >
template<typename B >
int array< T >::BinaryHeap< B >::parent (
            int i )  [inline]
```

Definition at line 105 of file main.cpp.

```
105                         {
106             return (i-1)/2;
107         }
```

3.2.2.5 remove()

```
template<typename T >
template<typename B >
B array< T >::BinaryHeap< B >::remove ( )  [inline]
```

Definition at line 128 of file main.cpp.

```
128                 {
129             B x = a[0];
130             a[0] = a[--n];
131             trickleDown(0);
132             if (3*n < a.length) resize();
133             return x;
134         }
```

3.2.2.6 right()

```
template<typename T >
template<typename B >
int array< T >::BinaryHeap< B >::right (
            int i )  [inline]
```

Definition at line 102 of file main.cpp.

```
102                     {
103             return 2*i + 2;
104         }
```

3.2.2.7 trickleDown()

```
template<typename T >
template<typename B >
void array< T >::BinaryHeap< B >::trickleDown (
            int i )  [inline]
```

Definition at line 136 of file main.cpp.

```
136                                 {
137
138             do {
139                 int j = -1;
140                 int r = right(i);
141                 if (r < n && compare(a[r], a[i]) < 0) {
```

```
142                     int l = left(i);
143                     if (compare(a[l], a[r]) < 0) {
144                         j = l;
145                         } else {
146                             j = r;
147                         }
148                     } else {
149                         int l = left(i);
150                         if (l < n && compare(a[l], a[i]) < 0) {
151                         j = l;
152                         }
153                     }
154             if (j >= 0) a.swap(i, j);
155             i = j;
156             } while (i >= 0);
157             }
```

3.2.3 Member Data Documentation

3.2.3.1 a

```
template<typename T >
template<typename B >
array<B> array< T >::BinaryHeap< B >::a
```

Definition at line 109 of file main.cpp.

3.2.3.2 n

```
template<typename T >
template<typename B >
int array< T >::BinaryHeap< B >::n
```

Definition at line 110 of file main.cpp.

The documentation for this struct was generated from the following file:

- /home/benson/CPTR227/Heaps/src/main.cpp

Chapter 4

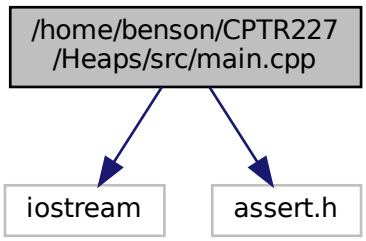
File Documentation

4.1 /home/benson/CPTR227/Heaps/src/main.cpp File Reference

This is a test of CMake, doxygen, and GitHub.

```
#include <iostream>
#include <assert.h>
```

Include dependency graph for main.cpp:



Classes

- struct array< T >
- struct array< T >::BinaryHeap< B >

Functions

- int main ()

4.1.1 Detailed Description

This is a test of CMake, doxygen, and GitHub.

This is the long brief at the top of main.cpp.

Author

Benson Nyakango

Date

1/28/2021

4.1.2 Function Documentation

4.1.2.1 main()

```
int main ( )
```

Definition at line 162 of file main.cpp.

```
162             {
163
164     BinaryHeap a;
165
166
167
168     return 0;
169 };
```

Index