

Inlab 6

Generated by Doxygen 1.9.6

| | |
|----------------------------------------------|----------|
| 1 Class Index | 1 |
| 1.1 Class List | 1 |
| 2 File Index | 3 |
| 2.1 File List | 3 |
| 3 Class Documentation | 5 |
| 3.1 BinarySearchTree Class Reference | 5 |
| 3.1.1 Detailed Description | 5 |
| 3.1.2 Constructor & Destructor Documentation | 5 |
| 3.1.2.1 BinarySearchTree() | 6 |
| 3.2 BSTNode Class Reference | 6 |
| 3.2.1 Detailed Description | 6 |
| 3.2.2 Constructor & Destructor Documentation | 6 |
| 3.2.2.1 BSTNode() | 6 |
| 3.3 DoublyLinkedList Class Reference | 7 |
| 3.3.1 Detailed Description | 7 |
| 3.3.2 Constructor & Destructor Documentation | 7 |
| 3.3.2.1 DoublyLinkedList() | 7 |
| 3.4 DoublyLinkedListNode Class Reference | 7 |
| 3.4.1 Detailed Description | 8 |
| 3.4.2 Constructor & Destructor Documentation | 8 |
| 3.4.2.1 DoublyLinkedListNode() [1/2] | 8 |
| 3.4.2.2 DoublyLinkedListNode() [2/2] | 8 |
| 3.5 Heap Class Reference | 8 |
| 3.5.1 Detailed Description | 9 |
| 3.5.2 Member Function Documentation | 9 |
| 3.5.2.1 deletemin() | 9 |
| 3.5.2.2 Heapify() | 9 |
| 3.5.2.3 insert() | 9 |
| 3.5.2.4 left() | 9 |
| 3.5.2.5 min() | 10 |
| 3.5.2.6 parent() | 10 |
| 3.5.2.7 right() | 10 |
| 3.6 SinglyLinkedList Class Reference | 10 |
| 3.6.1 Detailed Description | 11 |
| 3.6.2 Constructor & Destructor Documentation | 11 |
| 3.6.2.1 SinglyLinkedList() | 11 |
| 3.6.3 Member Function Documentation | 11 |
| 3.6.3.1 deleteVal() | 11 |
| 3.6.3.2 find() | 12 |
| 3.6.3.3 insert() | 12 |
| 3.6.3.4 printer() | 12 |

| | |
|----------------------------------------------|-----------|
| 3.6.3.5 reverse() | 12 |
| 3.7 SinglyLinkedListNode Class Reference | 12 |
| 3.7.1 Detailed Description | 13 |
| 3.7.2 Constructor & Destructor Documentation | 13 |
| 3.7.2.1 SinglyLinkedListNode() [1/2] | 13 |
| 3.7.2.2 SinglyLinkedListNode() [2/2] | 13 |
| 3.8 Trie Class Reference | 14 |
| 3.8.1 Detailed Description | 14 |
| 3.8.2 Constructor & Destructor Documentation | 14 |
| 3.8.2.1 Trie() | 14 |
| 4 File Documentation | 15 |
| 4.1 DSA.cpp File Reference | 15 |
| 4.1.1 Detailed Description | 16 |
| 4.1.2 Function Documentation | 16 |
| 4.1.2.1 merge() | 16 |
| Index | 17 |

Chapter 1

Class Index

1.1 Class List

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

| | | |
|--------------------------------------|--------------------------------------------------------------------------------------------------------------------|----|
| BinarySearchTree | Brief Description of BinarySearchTree . Brief Description continued | 5 |
| BSTNode | Brief Description of BSTNode . Brief Description continued | 6 |
| DoublyLinkedList | Brief Description of DoublyLinkedList . Brief Description continued | 7 |
| DoublyLinkedListNode | Brief Description of DoublyLinkedListNode . Brief Description continued | 7 |
| Heap | Brief Description of Heap . Used to do operations on Heap Data Structure | 8 |
| SinglyLinkedList | Brief Description about SinglyLinkedList | 10 |
| SinglyLinkedListNode | Brief Description of SinglyLinkedListNode . Brief Description continued | 12 |
| Trie | Brief Description of Trie . Brief Description continued | 14 |

Chapter 2

File Index

2.1 File List

Here is a list of all documented files with brief descriptions:

| | | |
|-------------------------|-------------------------------------------------|----|
| DSA.cpp | It is about various Linked List Nodes | 15 |
|-------------------------|-------------------------------------------------|----|

Chapter 3

Class Documentation

3.1 BinarySearchTree Class Reference

Brief Description of [BinarySearchTree](#). Brief Description continued.

Public Types

- enum **order** { **PRE** , **IN** , **POST** }

Public Member Functions

- [BinarySearchTree](#) ()
- void **insert** (ll val)
- void **traverse** ([BSTNode](#) *T, order tt)
- ll **height** ([BSTNode](#) *T)

Public Attributes

- [BSTNode](#) * **root**

3.1.1 Detailed Description

Brief Description of [BinarySearchTree](#). Brief Description continued.

Detailed description starts here.

3.1.2 Constructor & Destructor Documentation

3.1.2.1 BinarySearchTree()

```
BinarySearchTree::BinarySearchTree ( ) [inline]
```

This is a constructor

[BinarySearchTree](#) Functions- insert, traverse, height

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

- [DSA.cpp](#)

3.2 BSTNode Class Reference

Brief Description of [BSTNode](#). Brief Description continued.

Public Member Functions

- [BSTNode](#) (ll val)

Public Attributes

- ll **info**
- ll **level**
- [BSTNode](#) * **left**
- [BSTNode](#) * **right**

3.2.1 Detailed Description

Brief Description of [BSTNode](#). Brief Description continued.

Detailed description starts here.

3.2.2 Constructor & Destructor Documentation

3.2.2.1 BSTNode()

```
BSTNode::BSTNode (
    ll val ) [inline]
```

This is a constructor

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

- [DSA.cpp](#)

3.3 DoublyLinkedList Class Reference

Brief Description of [DoublyLinkedList](#). Brief Description continued.

Public Member Functions

- [DoublyLinkedList](#) ()
- void **insert** (ll data)
- void **printer** (string sep=", ")
- void **reverse** ()

Public Attributes

- [DoublyLinkedListNode](#) * **head**
- [DoublyLinkedListNode](#) * **tail**

3.3.1 Detailed Description

Brief Description of [DoublyLinkedList](#). Brief Description continued.

Detailed description starts here.

3.3.2 Constructor & Destructor Documentation

3.3.2.1 DoublyLinkedList()

```
DoublyLinkedList::DoublyLinkedList ( ) [inline]
```

This is a constructor

[DoublyLinkedListNode](#) functions- insert, printer, reverse

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

- [DSA.cpp](#)

3.4 DoublyLinkedListNode Class Reference

Brief Description of [DoublyLinkedListNode](#). Brief Description continued.

Public Member Functions

- [DoublyLinkedListNode](#) ()
- [DoublyLinkedListNode](#) (ll val)

Public Attributes

- `ll data`
- `DoublyLinkedListNode * next`
- `DoublyLinkedListNode * prev`

3.4.1 Detailed Description

Brief Description of [DoublyLinkedListNode](#). Brief Description continued.

Detailed description starts here.

3.4.2 Constructor & Destructor Documentation

3.4.2.1 DoublyLinkedListNode() [1/2]

```
DoublyLinkedListNode::DoublyLinkedListNode ( ) [inline]
```

This is a constructor

3.4.2.2 DoublyLinkedListNode() [2/2]

```
DoublyLinkedListNode::DoublyLinkedListNode (
    ll val ) [inline]
```

This is a constructor

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

- [DSA.cpp](#)

3.5 Heap Class Reference

Brief Description of [Heap](#). Used to do operations on [Heap](#) Data Structure.

Public Member Functions

- `int left (int parent)`
- `int right (int parent)`
- `int parent (int child)`
- `void Heapify (int index)`
- `void insert (int element)`
- `void deletemin ()`
- `int min ()`

3.5.1 Detailed Description

Brief Description of [Heap](#). Used to do operations on [Heap](#) Data Structure.

It has various functions like left, right, parent, Heapify, insert,deletemin,min

3.5.2 Member Function Documentation

3.5.2.1 deletemin()

```
void Heap::deletemin ( )
```

Brief Description of deletemin function.

It is used to delete the minimum value from [Heap](#)

3.5.2.2 Heapify()

```
void Heap::Heapify (
    int index )
```

Brief Description of Heapify function.

It is used to Heapify the elements from [Heap](#)

3.5.2.3 insert()

```
void Heap::insert (
    int element )
```

Brief Description of Insert function.

It is used to insert a element in [Heap](#).

3.5.2.4 left()

```
int Heap::left (
    int parent )
```

Brief Description of left function.

It is used to find the left element of child from [Heap](#)

3.5.2.5 min()

```
int Heap::min ( )
```

Brief Description of min function.

It is used to return the minimum value from [Heap](#)

3.5.2.6 parent()

```
int Heap::parent (
    int child )
```

Brief Description of parent function.

It is used to find the parent element from [Heap](#)

3.5.2.7 right()

```
int Heap::right (
    int parent )
```

Brief Description of right function.

It is used to find the right element of child from [Heap](#)

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

- [DSA.cpp](#)

3.6 SinglyLinkedList Class Reference

Brief Description about [SinglyLinkedList](#).

Public Member Functions

- [SinglyLinkedList](#) ()
This is about Singly Linked List.
- void [insert](#) (ll data)
- [SinglyLinkedListNode](#) * [find](#) (ll data)
- bool [deleteVal](#) (ll data)
- void [printer](#) (string sep=", ")
- void [reverse](#) ()

Public Attributes

- [SinglyLinkedListNode](#) * **head**
- [SinglyLinkedListNode](#) * **tail**

3.6.1 Detailed Description

Brief Description about [SinglyLinkedList](#).

Brief Description continued.

3.6.2 Constructor & Destructor Documentation

3.6.2.1 SinglyLinkedList()

```
SinglyLinkedList::SinglyLinkedList ( ) [inline]
```

This is about Singly Linked List.

Parameters

| | | |
|-----|-------------|--|
| out | <i>head</i> | |
|-----|-------------|--|

This is constructor w/o parameter

head - One of the variables used

tail - Another variable used

data - Another variable used

ptr - Another variable used

Functions- insert,find, deleteVal, printer, reverse

3.6.3 Member Function Documentation

3.6.3.1 deleteVal()

```
bool SinglyLinkedList::deleteVal (
    ll data ) [inline]
```

Parameters

| | | |
|----|-------------|--|
| in | <i>data</i> | |
|----|-------------|--|

This is member function deleteVal

3.6.3.2 find()

```
SinglyLinkedListNode * SinglyLinkedList::find (
    ll data ) [inline]
```

Parameters

| | | |
|-----|-------------|--|
| in | <i>data</i> | |
| out | <i>prev</i> | |

This is member function

3.6.3.3 insert()

```
void SinglyLinkedList::insert (
    ll data ) [inline]
```

Parameters

| | | |
|----|-------------|--|
| in | <i>data</i> | |
|----|-------------|--|

This is member function insert

3.6.3.4 printer()

```
void SinglyLinkedList::printer (
    string sep = ", " ) [inline]
```

This is member function printer

3.6.3.5 reverse()

```
void SinglyLinkedList::reverse ( ) [inline]
```

This is member function reverse

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

- [DSA.cpp](#)

3.7 SinglyLinkedListNode Class Reference

Brief Description of [SinglyLinkedListNode](#). Brief Description continued.

Public Member Functions

- [SinglyLinkedListNode](#) ()
- [SinglyLinkedListNode](#) (ll val)

Public Attributes

- ll **data**
This is about Singly Linked List Node.
- [SinglyLinkedListNode](#) * **next**

3.7.1 Detailed Description

Brief Description of [SinglyLinkedListNode](#). Brief Description continued.

Detailed description starts here.

3.7.2 Constructor & Destructor Documentation

3.7.2.1 SinglyLinkedListNode() [1/2]

```
SinglyLinkedListNode::SinglyLinkedListNode ( ) [inline]
```

Parameters

| | | |
|-----|-------------|--|
| out | <i>data</i> | |
|-----|-------------|--|

This is constructor w/o parameter

3.7.2.2 SinglyLinkedListNode() [2/2]

```
SinglyLinkedListNode::SinglyLinkedListNode (
    ll val ) [inline]
```

Parameters

| | | |
|-----|-------------|--|
| in | <i>val</i> | |
| out | <i>data</i> | |

This is destructor with parameter val

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

- [DSA.cpp](#)

3.8 Trie Class Reference

Brief Description of [Trie](#). Brief Description continued.

Public Member Functions

- [Trie](#) ()
- bool **find** ([Trie](#) *T, char c)
- void **insert** (string s)
- bool **checkPrefix** (string s)
- ll **countPrefix** (string s)

Public Attributes

- ll **count**
- map< char, [Trie](#) * > **nodes**

3.8.1 Detailed Description

Brief Description of [Trie](#). Brief Description continued.

Detailed description starts here.

3.8.2 Constructor & Destructor Documentation

3.8.2.1 [Trie](#)()

```
Trie::Trie ( ) [inline]
```

This is a constructor

[Trie](#) Functions- find, insert, checkPrefix, countPrefix

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

- [DSA.cpp](#)

Chapter 4

File Documentation

4.1 DSA.cpp File Reference

It is about various Linked List Nodes.

```
#include <bits/stdc++.h>
```

Classes

- class [SinglyLinkedListNode](#)
Brief Description of [SinglyLinkedListNode](#). Brief Description continued.
- class [SinglyLinkedList](#)
Brief Description about [SinglyLinkedList](#).
- class [DoublyLinkedListNode](#)
Brief Description of [DoublyLinkedListNode](#). Brief Description continued.
- class [DoublyLinkedList](#)
Brief Description of [DoublyLinkedList](#). Brief Description continued.
- class [BSTNode](#)
Brief Description of [BSTNode](#). Brief Description continued.
- class [BinarySearchTree](#)
Brief Description of [BinarySearchTree](#). Brief Description continued.
- class [Trie](#)
Brief Description of [Trie](#). Brief Description continued.
- class [Heap](#)
Brief Description of [Heap](#). Used to do operations on [Heap](#) Data Structure.

Macros

- `#define ll long long int`
- `#define vi vector<int>`
- `#define vll vector<ll>`

Functions

- ostream & **operator**<< (ostream &out, const [SinglyLinkedListNode](#) &node)
- [SinglyLinkedList](#) merge ([SinglyLinkedList](#) list1, [SinglyLinkedList](#) list2)
- ostream & **operator**<< (ostream &out, const [DoublyLinkedListNode](#) &node)
- ostream & **operator**<< (ostream &out, const [BSTNode](#) &node)
- int **main** ()

4.1.1 Detailed Description

It is about various Linked List Nodes.

Author

Yash Kulkarni

Date

Sep 21 2022

4.1.2 Function Documentation

4.1.2.1 merge()

```
SinglyLinkedList merge (  
    SinglyLinkedList list1,  
    SinglyLinkedList list2 )
```

Parameters

| | | |
|-----|---------------|--|
| in | <i>list1</i> | |
| in | <i>list2</i> | |
| out | <i>merged</i> | |

This is function merge

Index

- BinarySearchTree, [5](#)
 - BinarySearchTree, [5](#)
- BSTNode, [6](#)
 - BSTNode, [6](#)
- deletemin
 - Heap, [9](#)
- deleteVal
 - SinglyLinkedList, [11](#)
- DoublyLinkedList, [7](#)
 - DoublyLinkedList, [7](#)
- DoublyLinkedListNode, [7](#)
 - DoublyLinkedListNode, [8](#)
- DSA.cpp, [15](#)
 - merge, [16](#)
- find
 - SinglyLinkedList, [12](#)
- Heap, [8](#)
 - deletemin, [9](#)
 - Heapify, [9](#)
 - insert, [9](#)
 - left, [9](#)
 - min, [9](#)
 - parent, [10](#)
 - right, [10](#)
- Heapify
 - Heap, [9](#)
- insert
 - Heap, [9](#)
 - SinglyLinkedList, [12](#)
- left
 - Heap, [9](#)
- merge
 - DSA.cpp, [16](#)
- min
 - Heap, [9](#)
- parent
 - Heap, [10](#)
- printer
 - SinglyLinkedList, [12](#)
- reverse
 - SinglyLinkedList, [12](#)
- right
 - Heap, [10](#)
- SinglyLinkedList, [10](#)
 - deleteVal, [11](#)
 - find, [12](#)
 - insert, [12](#)
 - printer, [12](#)
 - reverse, [12](#)
 - SinglyLinkedList, [11](#)
- SinglyLinkedListNode, [12](#)
 - SinglyLinkedListNode, [13](#)
- Trie, [14](#)
 - Trie, [14](#)