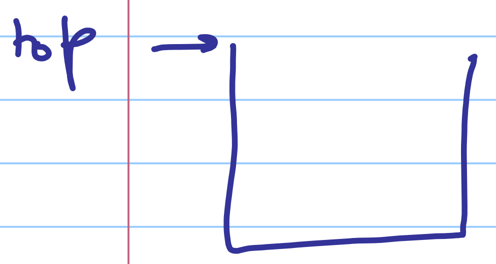
COL100 Lecture 21

Note Title 22-10-2018

Reminder 26th Oct.

Review:

Stack



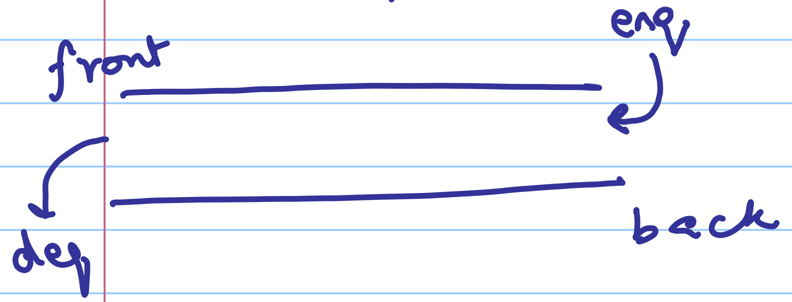
s.push (value)
value s.pop ()
s.size()
s. peek()
s. isEmpty()

LIFO stack

s.push(value)
void s.pop() does not return value
s.size ()
s.top()
s.empty()

Queue

FIFO



q. enqueue (value)
q. dequeue ()
q. peek ()
q. size()
q. isEmpty ()

queue

q.push (value)
void q.pop()
q. front()
q. size()
q.empty ()

Sets : ADTs for /Collection

Motivating example:
count the number of unique
words in some text.

e.g. - count unique visitors to a
website

Sets answer membership queries

```
     s.add (value)
     s.remove ( value)
bool s. contains (value)   : returns TRUE
                              if value in set s
```

Looping over Sets using for-each loop

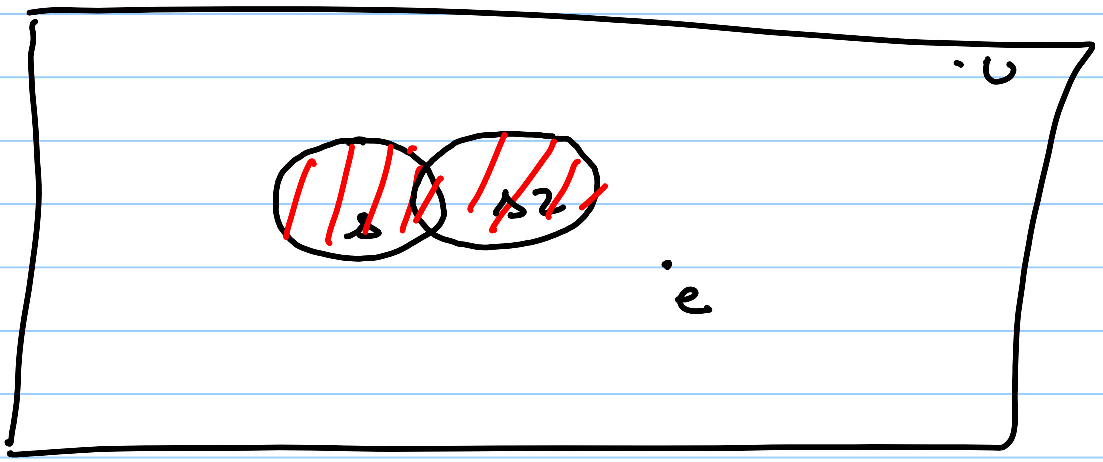
for

```
#include "set.h"          // <set>

set<string> friends;

friends.add("xyz");      // { "xyz" }
friends.add("abc");      // {"abc", "xyz"}
friends.add("xyz");      // {"abc", "xyz"}

for (string myfriend : friends)
{
        cout << "Hi " << myfriend
                                    << endl;
        break;
}
```

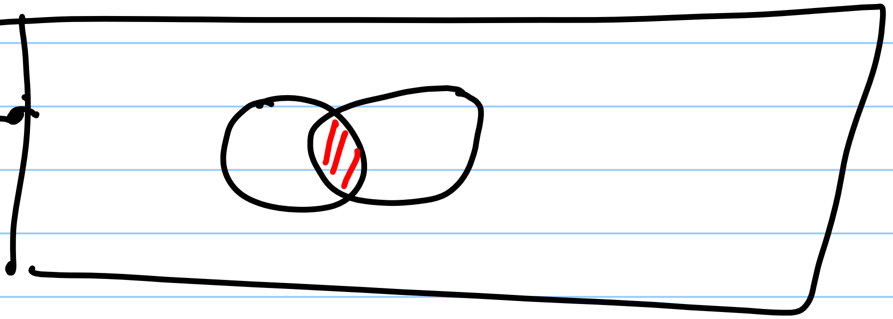
set maintained in lexicographic order of elements

Venn diagrams

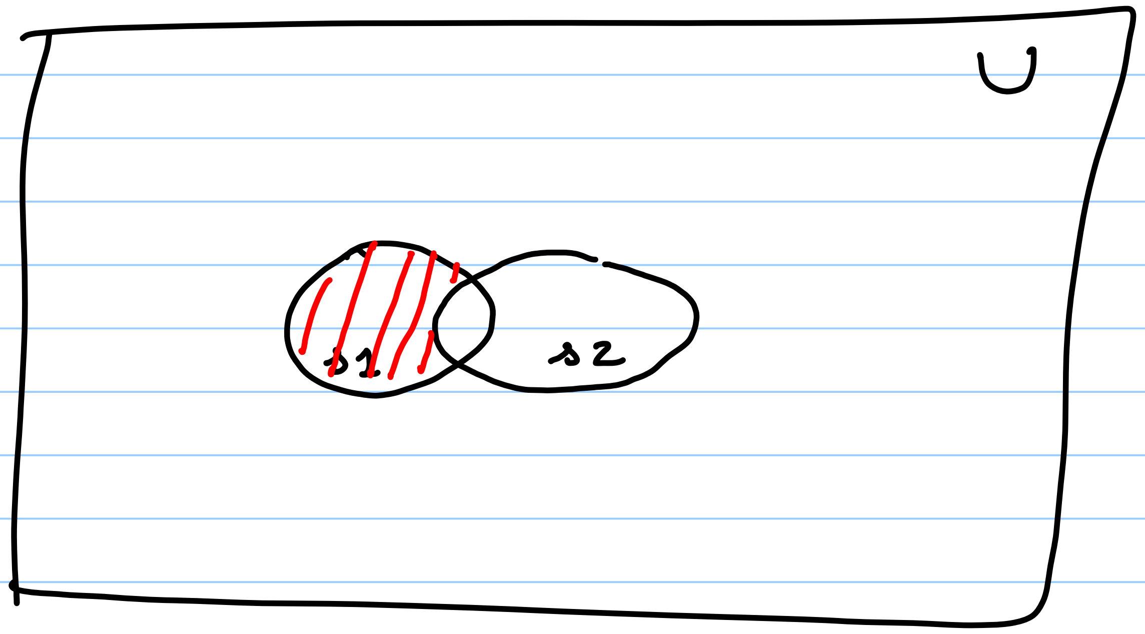
Set Union



Set Intersection



Set Difference

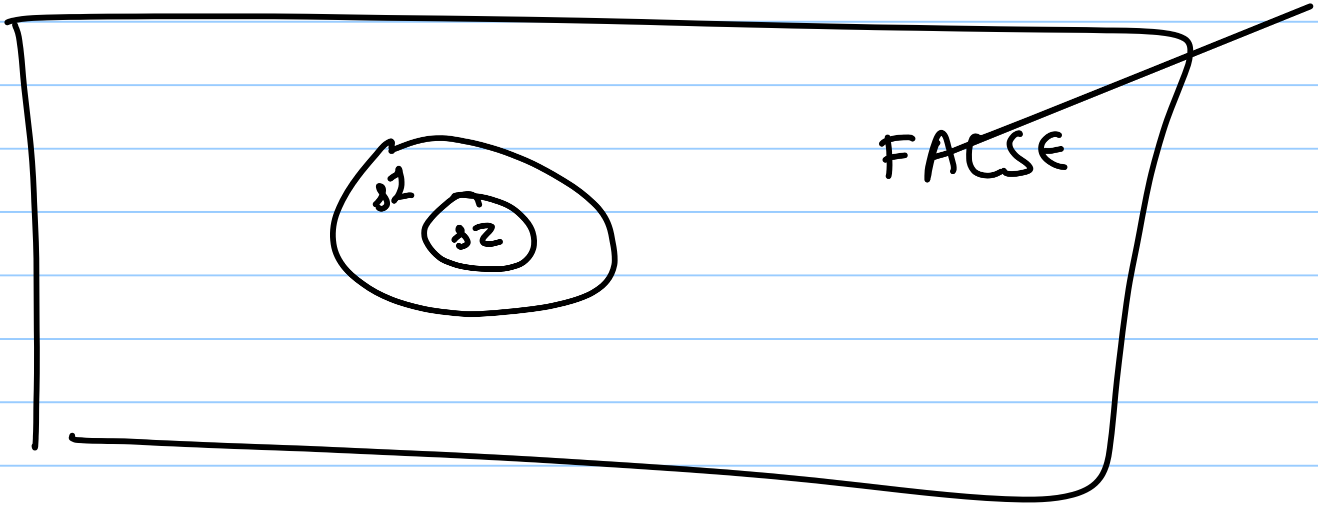


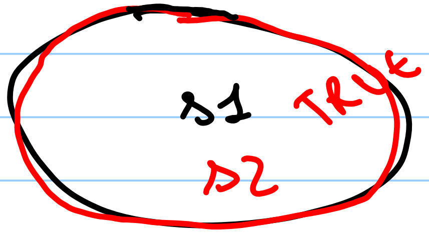
s1 - s2 ?

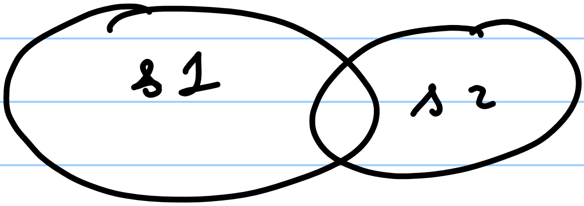
Member function

s.add (value)	s.insert(value)	O (log N)	adds an element to s if not already present
s.remove (value)	s.erase(value)	O (log N)	removes an element from s if it exists
s.contains (value)	s.find (value) != s.end()	O(log N)	returns true if value exists in set s
s.size()	s.size ()	O (1)	return size of s
s.isEmpty()	s.empty ()	O (1)	TRUE if empty
s. clear()	s.clear()	O (N)	removes all elements
s. first ()	* s.begin()	O (log N)	returns value of first element

s.equels(s2)	s == s2	O(N)	return true if the two sets s and s2 contain exactly the same elements
s.isSubsetOf(s2) s ⊆ s2 ?	No counterpart	O(N)	returns true if all elements present in s are also present in s2
s.toString()	No counterpart	O(N)	converts to string representati







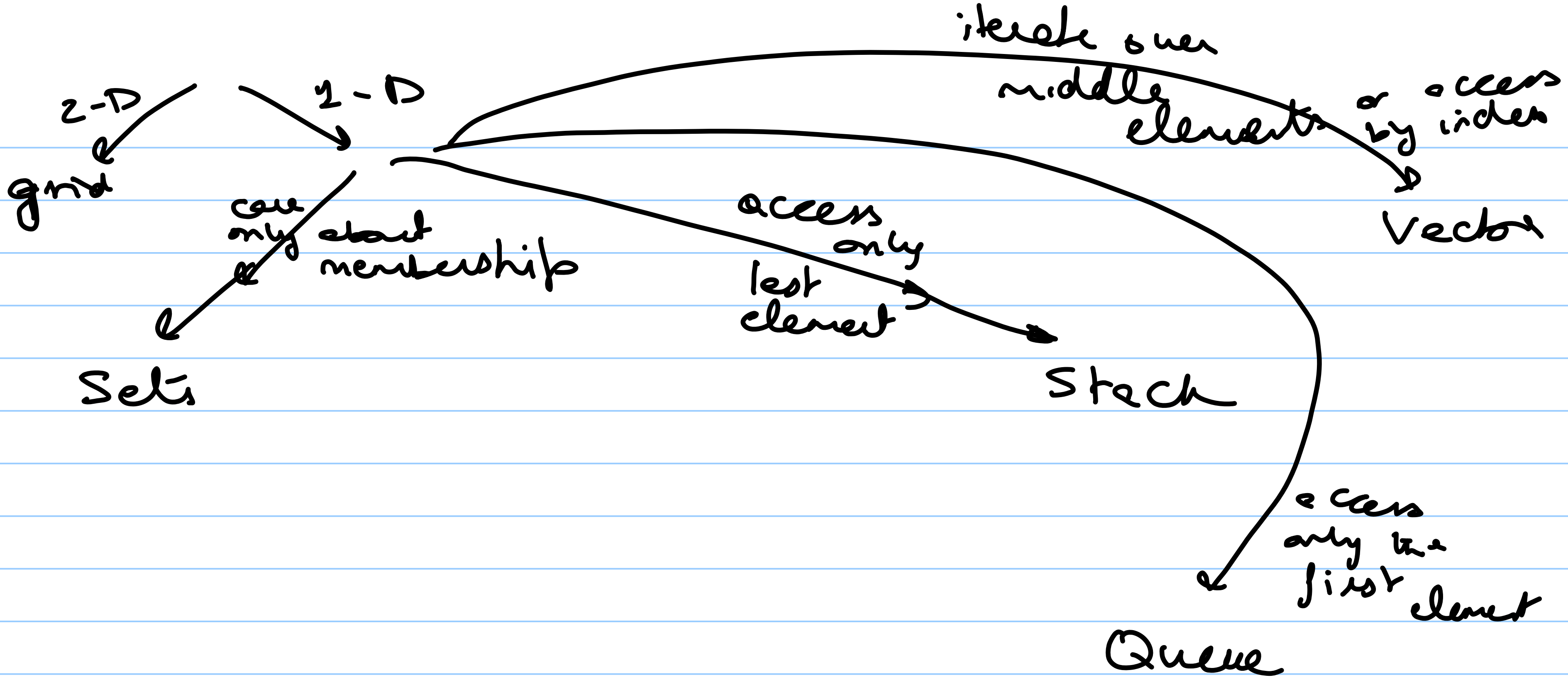
FALSE

Implementing **isSubsetOf()**

~~Set<string> s1, s2;~~

~~// populate s1, s2~~

```
bool    isSubsetOf( Set<string> const &s1,
                    set <string> const &s2 )
{
            for ( string  e : s1 )
O(          {
NlogN)  O(logN) ←   if ( s2.find(e) == s2.end() )
                    {
                        return false;
                    }
            }
            return true;
}
```


2-D
grid
1-D
care only about membership
Sets
access only last element
Stack
iterate over middle elements
or access by index
Vector
access only the first element
Queue

Nested ADTs

```
Set<Vector<string>>

v1:   { "abc" , "xyz" }
v2:   { "xyz" , "abc" }
v3:   { "abc" , "xyz" }
   s.add(v1);
   s.add(v2);
   s.add(v3);   // no effect.
```