

SCC120 Week 7 workshop answers

Set ADT and Stack ADT

1) The key concept of abstraction is to focus on the essential logic of a data collection and to ignore specific details of the elements it contains. Applying this to an ADT, we care about the key operations that can be performed on an ADT. The ADT does not need to know about the types of elements or how it is implemented (e.g. with an array or a linked list).

2) We discussed the four methods below, but there can also be others.

```
public class Set
{
    public Set();
    public void add(Element X);
    public void delete(Element X);
    public boolean member(Element X);
    public int size();
}
```

3) We don't need the "noElements" variable, since it holds the same value as the variable "i". And then we can just return "i" at the end.

4) We assume there is a "top" variable pointing to the first node in the linked list, and that the fields of each "node" in the linked list include a "value" and a "next" pointer. For exists(X):

```
p = top;
while (p != NULL) {
    if (p->value == X)
        return true;
    p = p->next;
}
return false;
```

For add(X), there are many ways to do this. Here is one example that uses exists(X):

```
if (!exists(X)) {
    node *p = new node();
    p->value = X;
    p->next = top;
    top = p;
}
```

The worst-case complexities in both cases are $O(n)$. Each method needs to iterate through each element in the worst case.

5) A Set ADT can be used to represent the set of courses you are taking (which have numbers such as SCC120 but can be considered to be unordered), or someone's friends in a social network. A Stack ADT is like a stack of plates, cups, or chairs in the real world.

6) Do this exercise yourself.

7) You call `push()` N times (N being the total number of numbers) to put each number onto a stack. And then you call `pop()` N times to get the list of numbers in reverse.

8) You call `pop()` N times to pop and read each number, while keeping track of the largest value so far with a variable. Then assuming you want to maintain the original stack, you keep track of the original set of numbers and call `push()` N times in the correct order to get back the original stack.