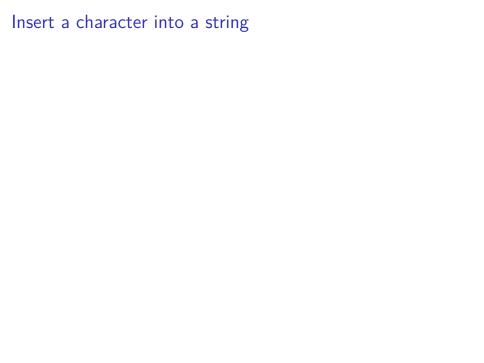
## Introduction to sorting

Dr Tom Ridge

2018 Q1



## A sorted string

Suppose s is a string of characters.

We say that s is *sorted* if the characters appear in ascending order.

For example, "abc" is sorted, but "cab" is not.

Challenge: insert

Given a **sorted** string s, and a character c, write a function

String insert(char c, String s)

This should insert c in the "correct" place according to the order. For example, insert('c', "abde") should produce the string "abcde".

#### Solution: insert

```
String insert(char c, String s) {
       String to_return = "";
       while(true) {
3
         if(s.equals("")) return to_return+c;
         char c2 = s.charAt(0);
5
         if(c>c2) { to return = to return + c2; }
6
         else { return to_return+c+s; }
         s=s.substring(1);
8
9
10
```

# Challenge: sort a string of characters

How can we use insert to sort a string s of characters?

- 1. Start with an empty string to\_return, which is trivially sorted
- 2. Look at the first character in s. Put it in the correct place in to\_return, using insert.
- 3. Drop the first character from s, and repeat from (1) until s is empty.

# Example execution

# Code: insertion sort on strings

```
String ins_sort(String s) {
String to_return = "";
while(true) {
   if(s.equals("")) return to_return;
   char c = s.charAt(0);
   to_return = insert(c,to_return);
   s=s.substring(1);
}
```



Can we generalize the code to sort a list of numbers?

#### Insert on integer lists

```
List insert(Integer i, List 10) {
1
       List to_return = nil();
2
       while(true) {
3
         if(l0.isEmpty()) return append1(to_return,i);
4
         Integer i2 = (Integer)(hd(10)); // note cast!!!
5
         if(i>i2) { to return = append1(to return,i2); }
6
         else { return append(append1(to return,i),10); }
7
         10=t1(10):
8
9
10
```

## Insertion sort for integer lists

```
List ins_sort(List 10) {

List to_return = nil();

while(true) {

if(10.isEmpty()) return to_return;

Integer i = (Integer)(hd(10)); // note cast!!!

to_return = insert(i,to_return);

10=tl(10);

}

}
```

#### Can we generalize further?

Clearly, the code for sorting a string of characters, and sorting a list of integers, is almost the same.

Can we write a version of insertion sort that works for any ordered collection of objects? (strings, lists, vectors, arrays etc)

We need the concept of an INTERFACE

#### A simple class

wrong (exception)

```
class Person {
  public String first=""; public String last="";
  Person(String f, String l) {first=f; last=l; }
  public String toString() { return first+" "+last; }
t = new Person("Tom", "Ridge");
u = new Person("Alf", "Bloggs");
Collections.sort(cons(t,cons(u,nil())));
What happens when you call the sort method? Something goes
```

#### Interfaces

See slides on "Java interfaces".

#### Recap

- We can sort things using insertion sort!
- ▶ We can sort strings of characters
- We can sort lists of integers
- But what happens if, say, we want to sort a list of people by first name?
- Or last name?
- Or age?
- ▶ Do we have to write the insertion sort code all over again each time? Fortunately not.

Recap: interfaces

What are they again?

#### Recap: string insertion sort

```
String insert(char c, String s) {
       String to_return = "";
       while(true) {
3
         if(s.equals("")) return to_return+c;
4
         char c2 = s.charAt(0);
5
         if(c > c2) { to return = to return + c2; }
6
         else { return to_return+c+s; }
7
         s=s.substring(1);
8
9
10
```

```
String ins_sort(String s) {
   String to_return = "";
   while(true) {
    if(s.equals("")) return to_return;
     char c = s.charAt(0);
   to_return = insert(c,to_return);
```

s=s.substring(1);

# Recap: integer list insertion sort

```
List insert(Integer i, List 10) {
1
       List to return = nil();
       while(true) {
3
         if(l0.isEmpty()) return append1(to_return,i);
         Integer i2 = (Integer)(hd(10)); // note cast!!!
5
         if(i>i2) { to_return = append1(to_return,i2); }
6
         else { return append(append1(to_return,i),10); }
         10=t1(10);
8
10
   (And similar for ins_sort)
```

#### Generalizing insertion sort

It is possible to generalize the insertion sort code

▶ to handle strings, lists, arrays, etc.

If you are interested, there are slides on "generic insertion sort"

but this material will not be examined in class tests.