Principles of Computer Science II

Sorting Algorithms

Ioannis Chatzigiannakis

Sapienza University of Rome

Lecture 5

Sorting in Python

```
1 list = [5, 6, 3, 7, 8, 11]
2 list.sort()
3
4 doubleList = [[8, 4, 5], [3, 8, 11], [4, 2, 19]]
5 doubleList.sort()
6
7
8 complexList = [('Alex', 5, M), ('Maria', 7, F), ('Katia',
       1, F), ('Bruno', 2, M), ('Artemis', 1, F)]
9 complexList.sort()
```

▶ How do we sort with different criteria?

Lambda Functions

▶ Lambda is a tool for building functions, or more precisely, for building function objects.
▶ Python has two tools for building functions: def and lambda.

Function declaration

```
1 def square_root(x): return math.sqrt(x)
2
3 def sum(x,y): return x + y
```

Lambda function

```
1 square_root = lambda x: math.sqrt(x)
2
3 sum = lambda x, y:  x + y
```

Lambda vs Functions

When using Lambda makes sense?
▶ the function is fairly simple, and
▶ it is going to be used only once.

When using Functions makes sense?
▶ to reduce code duplication, or

> If your application contains duplicate chunks of code in various places, then you can put one copy of that code into a function, and then call it from various places in your code.

▶ to modularize code.

> If you have a chunk of code that performs one well-defined operation but is really long and interrupts the readable flow of your program.

What Can be expressed using Lambda

- ▶ If it does not return a value, it is not an expression and cannot be put into a lambda.
- ▶ If you can imagine it in an assignment statement, on the right-hand side of the equals sign, it is an expression and can be put into a lambda.

What Can be expressed using Lambda

1. Assignment statements cannot be used in lambda – do not return anything, not even None (null).
2. Simple things: mathematical operations, string operations, list comprehensions, etc. are OK in a lambda.
3. Function calls are expressions. It is OK to put a function call in a lambda, and to pass arguments to that function. Doing this wraps the function call (arguments and all) inside a new, anonymous function.
4. In Python 3, print became a function, so in Python 3+, print(...) can be used in a lambda.
5. Functions that return None: like the print function in Python 3, can be used in a lambda.
6. Conditional expressions, return a value, and can be used in a lambda.

Lambda Examples

```
1 lambda: a if some_condition() else b
2
3 lambda x: 'big' if x > 100 else 'small'
4
5 out=lambda *x: print(" ".join(map(str,x)))
```

Data Assignment For Lists

Set an item in a list using the member function __setitem__

```
1 list[4] = 42
2 list.__setitem__(4,42)
```

Example: function that swaps two elements in a given list

```
1 def swap(a,x,y):
2     a[x] = (a[x], a[y])
3     a[y] = a[x][0]
4     a[x] = a[x][1]
```

Example: lambda expression that swaps two elements in a given list

```
1 swap = lambda a,x,y:(lambda f=a.__setitem__:
2         (f(x,(a[x],a[y])), f(y,a[x][0]), f(x,a[x][1])))()
```

Sorting with Lambda functions

```
1 doubleList = [[8, 4, 5], [3, 8, 11], [4, 2, 19], [3, 2,
      19]]
2 doubleList.sort()
3 doubleList.sort(key=lambda x: x[2])
4 doubleList.sort(key=lambda x: -x[0])
5 doubleList.sort(key=lambda x: (-x[0], x[1]))
```

Selection Sort Algorithm

This algorithm first finds the smallest element in the array and exchanges it with the element in the first position, then find the second smallest element and exchange it with the element in the second position, and continues in this way until the entire array is sorted.

Selection Sort: Example

Original Array	After 1st pass	After 2nd pass	After 3rd pass	After 4th pass	After 5th pass
3	1	1	1	1	1
6	6	3	3	3	3
1	3	6	4	4	4
8	8	8	8	5	5
4	4	4	6	6	6
5	5	5	5	8	8

Selection Sort Code

```
1  a = [5, 1, 6, 2, 4, 3]
2  for i in range(0, len(a)):
3      min = i
4      for j in range(i + 1, len(a) - 1):
5          if a[j] > a[min]:
6              min = j
7
8      temp = a[i]
9      a[j] = a[min]
10     a[min] = temp
```

How good is Selection Sort?

- How many comparisons are required until the list is sorted?

How good is Selection Sort?

- How many comparisons are required until the list is sorted?
  - 1st loop: n - 1
  - 2nd loop: n - 2
  - ...

How good is Selection Sort?

- How many comparisons are required until the list is sorted?
  - 1st loop: n - 1
  - 2nd loop: n - 2
  - ...
  - (n-1)+(n-2)+(n-3)+ ...+3+2+1 comparisons are required

How good is Selection Sort?

- How many comparisons are required until the list is sorted?
  - 1st loop: n - 1
  - 2nd loop: n - 2
  - ...
  - (n-1)+(n-2)+(n-3)+ ...+3+2+1 comparisons are required
  - $\sum \frac{n(n-1)}{2}$ comparisons are required

How good is Selection Sort?

- How many comparisons are required until the list is sorted?
  - 1st loop: n - 1
  - 2nd loop: n - 2
  - ...
  - (n-1)+(n-2)+(n-3)+ ...+3+2+1 comparisons are required
  - $\sum \frac{n(n-1)}{2}$ comparisons are required
- How much memory is needed ?

How good is Selection Sort?

- How many comparisons are required until the list is sorted?
  - 1st loop: n - 1
  - 2nd loop: n - 2
  - ...
  - (n-1)+(n-2)+(n-3)+ ...+3+2+1 comparisons are required
  - $\sum \frac{n(n-1)}{2}$ comparisons are required
- How much memory is needed ?
  - 1 additional slot.