Efficiency continued

Arnamoy Bhattacharyya

Agenda

Revisit example from last lecture

Bubble Sort efficiency

Exercise

Efficiency

How does the running of this code fragment depend on **n**?

```
i, sum = 0, 0
while i**2 < n:
    j = 0
    while j**2 < n:
        sum += i * j
        j += 1
    i += 1
```

Empirical estimation

Bubble Sort efficiency

Basic sorting algorithm

Very inefficient

Not practical to use for very long array

NO recursion

Bubble Sort

```
def bubble_sort(list_: list) -> list:
    """
    Sort the items in list_ in non-decreasing order.

    """
    j = len(list_) - 1
    while j != 0:
        # Swap every item that is out of order.
        for i in range(j):
            if list_[i] > list_[i + 1]:
                list_[i], list_[i + 1] = list_[i + 1], list_[i]
        j -= 1
    return list_
```

In Class Exercises