

3.3 Complexity of Algorithms

3.3 pg 229 # 1

Give a big- O estimate for the number of operations (where an operation is an addition or a multiplication) used in this segment of an algorithm.

```
t := 0
for i := 1 to 3
  for j := 1 to 4
    t := t + ij
```

3.3 pg 229 # 3

Give a big- O estimate for the number of operations, where an operation is a comparison or a multiplication, used in this segment of an algorithm (ignoring comparisons used to test the conditions in the for loops, where a_1, a_2, \dots, a_n are positive real numbers).

```
m := 0
for i := 1 to n
  for j := i + 1 to n
    m := max( $a_i a_j$ , m)
```

3.3 pg 230 #21

What is the effect in the time required to solve a problem when you increase the size of the input from n to $n + 1$, assuming that the number of milliseconds the algorithm used to solve the problem with input size n is each of these function? [Express your answer in the simplest form possible, either as a ratio or a difference. Your answer may be a function of n or a constant.]

- a) $\log n$
- b) $100n$
- c) n^2
- d) n^3
- e) 2^n
- g) $n!$