Algorithm efficiency examples

CS110C

Max Luttrell, CCSF

execution time

- example: print out every item in a linked list with ***n*** items

```
1 Node<ItemType>* curPtr = headPtr;
2 while (curPtr != nullPtr)
3 {
4   cout << curPtr->getItem() << endl;
5   curPtr = curPtr->getNext();
6 }
```

```
Line 1: 1 assignment
```

```
Line 2: n+1 comparisons
```

```
Line 4: n outputs
```

```
Line 5: n assignments
```

```
total operations: (n+1)*a + (n+1)*c + n*o
a: cost of assignment
c: cost of comparison
o: cost of output
```

execution time

- example: nested loop

```
for (i = 1 through n)
  for (j = 1 through n)
    for (k = 1 through n)
      task T
```

- Example: task T takes 1 second
- **inner loop: *n * 1 = n seconds***

execution time

- example: nested loop

```
for (i = 1 through n)
  for (j = 1 through n)
    for (k = 1 through n)
      task T
```

- Example: task T takes 1 second
- inner loop: n * 1 = n seconds
- **loop on j: n * n seconds**

execution time

- example: nested loop

```
for (i = 1 through n)
  for (j = 1 through n)
    for (k = 1 through n)
      task T
```

- Example: task T takes 1 second
- inner loop: n * 1 = n seconds
- loop on j: n * n seconds
- **loop on i: n * n * n seconds**

- In general, if we assume task T takes t time units:
- inner loop: n * _t_
- loop on j: (n * _t_) * _n_
- loop on i: ((_n_ * _t_) * _n_)) * _n_

execution time

- example: nested loop

```
for (i = 1 through n)
  for (j = 1 through i)
    for (k = 1 through 5)
      task T
```

- assume task T takes t time units.
- inner loop: *5 * t*
- loop on j: *5 * t * i*
- loop on i: $\sum_{i=1}^{n}(5*t*i)$ = *(5 * t) * (1 + 2 + … + n) = 5 * t * n * (n+1) / 2*

Math formula:

$$\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$$