CSC 165

program lower bound

week 11, lecture 1

Danny Heap

heap@cs.toronto.edu

www.cdf.toronto.edu/~heap/165/F09

counting minimum costs

```
def IS(A) :
    """ IS(A) sorts the elements of A in non-decreasing order """
    i = 1
    while i < len(A) :
        t = A[i]
        j = i
        while j > 0 and A[j-1] > t :
            A[j] = A[j-1] # shift up
            j = j-1
        A[j] = t
        i = i+1
```

I want to prove that $W_{\text{IS}} \in \Omega(n^2)$.

I want to choose a badly-performing (though perhaps not the worst) example

scratch

computer mis-statements

You will have encountered unsatisfying results in python:

- ```
  >>> x = 1/10.0
  >>> x
  >>> for i in range(9) : x += 1/10.0
  >>> x
  ```
- ```
  >>> import math
  >>> math.pi
  >>> math.e
  ```
- ```
  >>> bf = 2.0
  >>> for i in range(10) :
  ...         bf *= bf
  ...         print bf
  ...
  ```
  ```

how numbers are represented

There are other ways to represent numbers: do arithmetic on ratios (scheme, lisp)
but there are always costs

If you fix the cost of arithmetic operations, you fix the size of numbers
Each number is given the same space (usually bits)

Result: floating numbers are represented in scientific notation using some base β,
a fixed number of digits, t, a certain range of exponents $e \in [e_{\min}, e_{\max}]$,
and some way to store the sign.

example

Suppose your base $\beta = 2$, you allow a bit for the sign,
you have room for $t = 3$ digits, and your exponents are from $[-2, 3]$.

There are several was to represent $1\frac{1}{2}$: 1.1×2^0,
0.11×2^1, 11.0×2^{-1}. Choose the normalized form —
There is one digit to the left of the radix point for non-zero quantities

What's the smallest positive number you can represent in this system?
What's the largest positive number you can represent in this system?

a number list

A number-line of the entire list of positive numbers isn't evenly-spaced

However the ratio of the gaps to the magnitude is roughly constant