# CSC 165

program examples

week 10, lecture 3

Danny Heap

heap@cs.toronto.edu

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

E3: posted, short, due Sunday
(yes, there is a grace day) (oh no! it must be typed!)

A3, E4: posted this weekend

# counting costs

want a coarse comparison of algorithms "speed" that ignores hardware, programmer virtuosity

compare some 'thing'

which speed do we care about: best, worst, average?
why?

define idealized "step" that doesn't depend on particular hardware and idealized "time" that counts the number of steps for a given input.

# linear search

1. $i = 0$ —
2. $0 < 4$ —
3. if $2 == 4$
5. $i = 1$
2. $1 < 4$
3. $4 == 4$
4. return 1

```
  def LS(A,x) :
    """ Return index i such that x == L[i].  Otherwise, return -1 """
1.     i = 0
2.     while i < len(A) :
3.         if A[i] == x :
4.             return i
5.         i = i + 1  -
6.     return -1
```

$4 = n$

Trace LS([2,4,6,8],4), and count the time complexity $t_{\text{LS}}([2,4,6,8],4)$

line 1 —  lines 2,3,5 $(0 \ldots j-1)$ — $j$ times  $3(j+1) + 1$

2,3,4 — 1 times

What is $t_{\text{LS}}(A, x)$, if the first index where $x$ is found is $j$?

line 1 — once  2,3,

What is $t_{\text{LS}}(A, x)$ is $x$ isn't in $A$ at all?

$3n$

line 1 — once  2,3,5 — $n$ times  0 special case?

line 2,6 once.

$3(n+1)$

# worst case

denote the worst-case complexity for program $P$ with input $x \in I$,
where the input size of $x$ is $n$ as
$W_P(n) = \max\{t_P(x) \mid x \in I \wedge \text{size}(x) = n\}$

size(x) = n

The upper bound $W_P \in \mathcal{O}(U)$ means

$$\exists c \in \mathbb{R}^+, \exists B \in \mathbb{N}, \forall n \in \mathbb{N}, n \geq B \;\Rightarrow\; \max\{t_P(x) \mid x \in I \wedge \text{size}(x) = n\} \leq cU(n)$$

That is: $$\exists c \in \mathbb{R}^+, \exists B \in \mathbb{N}, \forall x \in I, \text{size}(x) \geq B \;\Rightarrow\; \underline{t_P(x) \leq cU(\text{size}(x))}$$

The lower bound $W_P \in \Omega(L)$ means

$$\exists c \in \mathbb{R}^+, \exists B \in \mathbb{N}, \forall n \in \mathbb{N}, n \geq B \;\Rightarrow\; \max\{t_P(x) \mid x \in I \wedge \text{size}(x) = n\} \geq cU(n)$$

That is: $$\exists c \in \mathbb{R}^+, \exists B \in \mathbb{N}, \forall n \in \mathbb{N}, n \geq B \;\Rightarrow\; \exists x \in I, \text{size}(x) = n \wedge t_P(x) \geq cL(n)$$

# bounding a sort

```
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
```

1

for each $i$? $j$: $n-1 \ldots 1$

$n$

$(n-1)$

# execute? – ~~2~~, 3, 4, 9, 8 — $n-1 \to (5 + 3n)$

$+ 1 + 1$

I want to prove that $W_{\text{IS}} \in \mathcal{O}(n^2)$. 1

# scratch

Set $C = \underline{\quad 10 \quad}$. Then $C \in \mathbb{R}^+$
Set $B = \underline{\quad 1 \quad}$. Then $B \in \mathbb{N}$.
Assume $A$ is an arbitrary array. Set $len(A) = n$ # convenience
assume $n \geq B$
Then lines 5-7 execute <u>at most</u> $n-1$ times,
plus 1 step for loop to fail: $3(n-1)+1$ times
$\leq 3n$

Then lines 2-9 contribute $n(5+3n)+1$ )
then line contribute <u>at most</u> 1
Thus $t_{IS}(A) \leq 3n^2+5n+2 \leq 10n^2 = Cn^2$ # $C=10$, $B=1$
Conclude $\exists C \in \mathbb{R}^+, \exists B \in \mathbb{N}, \forall A \in I$, $size(A) \geq B$
$\Longrightarrow t_{IS}(A) \leq cn^2$

So $W_{IS} \in \Theta(n^2)$