office hours M : 3-4:30 F: 1:30-3:30
W : 2-4

CSC165 fall 2014

Mathematical expression

Danny Heap
heap@cs.toronto.edu
BA4270 (behind elevators)
http://www.cdf.toronto.edu/~heap/165/F14/
416-978-5899

Course notes, chapter 4

Outline

more asymptotics

notes

counting costs

resource: time

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

165
236
263

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

idealized steps

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

list, len(A) = n

value

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

1

n +1

n

x

n

1

oops, wasn't there

1 time

j+1

j+1

1

j

$\boxed{3j+4}$

$3(n-1)+4$

$3n-3+4 = \underline{3n+1}$

$10 = 3\cdot 2 + 4$

worst case $3n+3$, $3(n+1)$

Trace LS([2,4,6,8],4), and count the time complexity

$t_{\mathrm{LS}}([2,4,6,8],4) = \underline{7}$

2

assume

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

$0 \le j < n$

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

worst case

perhaps

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

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

worst case no worse than this

break-point.

$\exists c \in \mathbb{R}^+, \exists B \in \mathbb{N}, \forall n \in \mathbb{N}, n \geq B$

multiplier

$\Rightarrow \max\{t_P(x) \mid x \in I \wedge \text{size}(x) = n\} \leq c\,U(n)$

That is: $\exists c \in \mathbb{R}^+, \exists B \in \mathbb{N}, \forall x \in I, \underbrace{\text{size}(x)}_{n} \geq B$

$\Rightarrow t_P(x) \leq c\,U(\underbrace{\text{size}(x)}_{n})$

eg $U(n) = n^2$

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

worst case. at least as bad as this

$\exists c \in \mathbb{R}^+, \exists B \in \mathbb{N}, \forall n \in \mathbb{N}, n \geq B$

$\Rightarrow \max\{t_P(x) \mid x \in I \wedge \text{size}(x) = n\} \geq cL(n)$

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

Insertion sort

list, $len(A) = n$

most work if A sorted greatest → smallest

[3, 1, 2]
[1, 3, 2]
[1, 2, 3]

```
def IS(A) :
    """ IS(A) sorts the elements of A in non-decreasing orc
    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 step

$i = 1, 2, \ldots, n-1$, $+1$

$n-1$

$n-1$

overestimate $j = i, i-1, \ldots, 1$, $+1$

$i-1$

$i-1$

$n-1$

$n-1$

$\underbrace{(2,3,4,8,9)}(n-1) + 1 + 1 + n-1$

$i < n$

$5(n-1) + 3(n-1)(i) + 2$

$\leq 5n + 3(n)(n) + n + 2$

$= 3n^2 + 6n + 2 \leq 11n^2$

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

big-oh of n^2

We know, or have heard, that all quadratic functions grow at "roughly" the same speed. Here's how we make "roughly" explicit.

$f(n) = 3 \in \Theta(n^2)$

$$\mathcal{O}(n^2) = \{f : \underbrace{\mathbb{N}}_{\text{input}} \mapsto \underbrace{\mathbb{R}^{\geq 0}}_{\text{output}} \mid \overbrace{\exists c \in \mathbb{R}^+}^{\text{mult.}}, \overbrace{\exists B \in \mathbb{N}}^{\text{breakpoint}}, \forall n \in \mathbb{N}, n \geq B \Rightarrow f(n) \leq cn^2\}$$

Those are a lot of symbols to process. They say that $\mathcal{O}(n^2)$ is a set of functions that take natural numbers as input and produce non-negative real numbers as output. An additional property of these functions is that for each of them you can find a multiplier c, and a breakpoint B, so that if you go far enough to the right (beyond B) the function is bounded above by cn^2.

In terms of limits, this says that as n approaches infinity, $f(n)$ is no bigger than cn^2 (once you find the appropriate c).

prove $W_{IS} \in \mathcal{O}(n^2)$ i.e. $\exists c \in \mathbb{R}^+, \exists B \in \mathbb{N}, (\forall n \in \mathbb{N}, n \geq B \Rightarrow W_{IS}(n) \leq c \cdot n^2)$

Choose $c =$ 11. Then $c \in \mathbb{R}^+$

Pick $B =$ 1. Then $B \in \mathbb{N}$

Assume $n \in \mathbb{N}$

Assume $n \geq B$

Assume A is a typical list, with $len(A) = n$.

Then any overestimate of $t_{IS}(A)$ will be an overestimate of $W_{IS}(n)$

squeeze next slide in here.

Then $n \geq B \Rightarrow W_{IS}(n) \leq c \cdot n^2$

Then $\forall n \in \mathbb{N}$, " "

Then $\exists B \in \mathbb{N}$, " "

Then $\exists c \in \mathbb{R}^+$, (")

prove $W_{\mathrm{IS}} \in \mathcal{O}(n^2)$

Then $t_{IS}(A)$ has

- line 1 costs $\leq \boxed{1}$ step $= 2$
- lines 2,3,4,8,9 cost $\boxed{1}$ step for each i in $\{1, 2, \ldots, n-1\}$, $\leq (n-1) \cdot 5$ steps
- lines 5,6,7 cost ≤ 1 step $+1$ step each for each j in $\{i, i-1, \ldots, 1\}$ $\leq (3 \cdot i + 1) \cdot (n-1)$

Summing up $t_{IS}(A) \leq 2 + 5(n-1) + (n-1)(3i+1)$

$\leq 2 + 5n + (3i+1)n$ # $n \geq n-1$

$\leq 2 + 5n + (3n+1)n$ # $n > n-1 \geq i$

$= 3n^2 + n + 5n + 2 = 3n^2 + 6n + 2$

$\leq 3n^2 + 6n \cdot n + 2 \cdot n^2$ # $n \geq 1$

$= 11n^2$

Then $t_{IS}(A) \leq cn^2$ # $c = 11$

Then $W_{IS}(n) \leq cn^2$ # since A chosen arbitrarily prove $W_{IS} \in \Omega(n^2)$ $\exists c \in \mathbb{R}^+, \exists B \in \mathbb{N}, \forall n \in \mathbb{N}, n \geq B \Rightarrow W_{IS}(n) \geq cn^2$

Proof

Pick $c = \underline{\quad 1/2 \quad}$. Then $c \in \mathbb{R}^+$

Pick $B = \underline{\quad 2 \quad}$. Then $B \in \mathbb{N}$

Assume $n \in \mathbb{N}$

Assume $n \geq B$

Choose $A = [n, n-1, \ldots, 1]$. Now lines 5, 6, 7 of $IS(A)$ must execute for $j = i, i-1, \ldots, 1$, since $A[i] = t$ is always less than its predecessors.

So, $t_{IS}(A) \geq 3i + 1$ for i in $\underbrace{\{1, \ldots, n-1\}}$

loop guard

$= 3(1)+1 + 3(2)+1 + \ldots + 3(n-1)+1$

$3(1+2+3+\ldots+n-1) + (n-1)$

Then $n \geq B \Rightarrow W_{IS}(n) \geq c \cdot n^2$

Then $\forall n \in \mathbb{N}$, " "

Then $\exists B \in \mathbb{N}$, " "

Then $\exists c \in \mathbb{R}^+$, " "

prove $W_{\mathrm{IS}} \in \Omega(n^2)$

$\rightarrow \; = 3(1 + \cdots + n-1) + n-1$

$= 3\,\dfrac{n(n-1)}{2} + n-1$

$= \dfrac{3n^2-3n}{2} + \dfrac{2(n-1)}{2}$

$= \dfrac{3n^2-n-2}{2} = \dfrac{n^2 + \overbrace{(n^2-n)}^{\geq 0} + \overbrace{(n^2-2)}^{\geq 0}}{2}$

$\geq \dfrac{n^2}{2}$ # $n \geq 2$

$= c \cdot n^2$ # $c = 1/2$

$t_{IS}(A) \geq c\,n^2$

Then $W_{IS}(n) \geq t_{IS}(A) \geq c\,n^2$

$1 + 2 + \cdots + n-1$

$n-1 + n-2 + \cdots + 2 + 1$

$n + n + \cdots + n + n$

$= (n-1)n$, counted twice, $S = \dfrac{n(n-1)}{2}$

maximum slice

```
def max_sum(L) :
  """maximum sum over slices of L"""
  max = 0
  i = 0
  while i < len(L) :
    j = i + 1
    while j <= len(L) :
      sum = 0
      k = i
      while k < j :
        sum = sum + L[k]
        k = k + 1
      if sum > max :
        max = sum
      j = j + 1
    i = i + 1
  return max
```

Notes

annotated slides

- monday's annotated slides
- wednesday's annotated slides