1/24/17 ⓪

TWO VOLUNTEER OPPORTUNITIES

(1) UCF H.S. Programming Tournament - **HOSTS**

3/16/2017 (Thurs) (7:45am - 5pm)

3/15/2017 (Wed 6-9 pm)

Steve ~~Ziens~~ Zielinski (s.zielinski72@gmail.)

(2) Junior Knights 10am - 1pm

Jan 28, Feb 4, 11, 18, Mar 4

Apr 1, 8, 15, 22, ~~25~~ 29

dmarino@cs.ucf.edu

(1) Name

(2) Date(s) you'd like to volunteer

(3) # of times

Finished dynamic memory allocation

Tools for analyzing the efficiency of algorithms. (Base Conversion)

If the "theoretical" run-time of an algorithm is known, we can make some predictions about how long it will take to run for different inputs.

If an algorithm runs in $O(f(n))$ for an input of size n that means that ~~we~~

$$\lim_{n\to\infty} \frac{T(n)}{f(n)} \leq c,$$ if $T(n)$ is the actual run time of the function, for some constant c.

for example, if $T(n) = 8n^2 - 6n + 4$

$$\lim_{n\to\infty} \frac{8n^2-6n+4}{n^2} = 8 \text{ (this is constant)}$$

thus, $T(n) = O(n^2)$.

$T(n) = O(n^3)$ because ✓ $\lim_{n\to\infty} \frac{8n^2-6n+4}{n^3} = 0$

$T(n) \neq O(n)$ $\lim_{n\to\infty} \frac{8n^2-6n+4}{n} = \infty$

If I tell you an algorithm runs in $O(f(n))$ time, then assume we can model its run time as follows: $T(n) = cf(n)$, for some const c.
(on input of size n)

Algorithm A sorts n numbers in $O(n^2)$ time. It takes 20 ms to sort 10,000 numbers. How long will it take to sort 40,000 numbers?

Let $T(n)$ be the actual run time of the alg.

$T(n) = cn^2$ for some const c.

$T(10000) = c(10^4)^2 = 20 \text{ ms}$

$c = \frac{20}{10^8} \text{ ms} = \frac{2}{10^7} \text{ ms}$

$T(40000) = \left(\frac{2}{10^7} \text{ ms}\right) \times 40000^2 = \frac{2}{\cancel{10^7}} \times 4^2 \times \cancel{10^8}^{\,10} \text{ ms}$

$= \boxed{320 \text{ ms}}$

A search of a database with n elements takes $O(\log_2 n)$ time. If 100,000 searches in a database of size 2^{16} takes 40 ms, how long will 600,000 searches take on a database with 2^{20} elements?

$T(n) = c \log_2 n$

$100{,}000 \times T(2^{16}) = c \log_2 2^{16} \times 100{,}000 = 40 \text{ ms}$

$c \cdot 16 \times 10^5 = 40 \text{ ms}$

$c = \frac{40}{16 \times 10^5} \text{ ms}$

$600{,}000 \times T(2^{20}) = 600{,}000 \times \frac{40}{16 \times 10^5} \text{ ms} \times \log_2 2^{20}$

$= \cancel{600{,}000}^{\,6} \times \frac{\cancel{40}^{\,10}}{4\,\cancel{16 \times 10^5}} \times 20 \text{ ms} = \boxed{300 \text{ ms}}$

1/24/17 3

An algorithm processing an $n \times m$ array takes $O(n^2 m^3)$ time (yeah it's really slow). On an array of size 100×200, the algorithm takes 1 second. How long would it take on an array sized 400×300?

$$T(n,m) = cn^2m^3$$
$$T(100,200) = c\,100^2 \cdot 200^3 = 1 \text{ sec}$$
$$= c\,8\times10^{10} = 1 \text{ sec}$$
$$c = \frac{1}{8\times10^{10}} \text{ sec}$$

NOTE: SKIPPED STEPS, PLEASE WORK OUT.

$$T(400,300) = \frac{1 \text{ sec}}{8\times10^{10}} \times 400^2 \times 300^3$$
$$= \frac{1 \text{ sec}}{8\times10^{10}} \times (4\times100)^2 \times (3\times100)^3$$
$$= \frac{1 \text{ sec}}{8\times10^{10}} \times 4^2 \times 100^2 \times 3^3 \times 100^3$$
$$= \frac{2\cancel{16}\times27}{\cancel{8\times10^{10}}} \times \cancel{10^{10}} \text{ sec}$$
$$= \boxed{54 \text{ seconds}}$$

WORKED OUT HERE...

$n = 30{,}000 \rightarrow t = 4 \quad (4.5)$

$n = 60{,}000 \rightarrow$

$$T(n) = cn^2$$
$$T(30000) = c\,30000^2 = 4.5 \text{ sec} \quad c = \frac{4.5}{30000^2} \text{ sec}$$
$$T(60000) = \frac{4.5 \text{ sec}}{30000^2} \times 60000^2 = (4.5 \text{ sec}) \times \left(\frac{60000}{30000}\right)^2$$
$$= 4.5 \text{ sec} \times 2^2 = 18 \text{ sec}$$

1/24/2017 4

$c(n + \frac{1}{2}n^2)$ $n = 30{,}000$

$30{,}000 + \frac{1}{2} \times \cancel{30{,}000^2}$

$15{,}000 \times 30{,}000$

Int overflows after ~ 2.1 billion

longlong goes to 4×10^{18}

1/24/17 (5)

Base Conversion

Our counting system is base 10:

$$356 = 3\times10^2 + 5\times10^1 + 6\times10^0$$

In general in base b, if we have a number

$$d_{n-1}d_{n-2}d_{n-3}\ldots d_0 = d_{n-1}\times b^{n-1} + d_{n-2}\times b^{n-2} + \cdots + d_0\times b^0$$

$$= \sum_{i=0}^{n-1} d_i \times b^i$$

Plug in $i=0$
then $i=1$ add all of these
then $i=2$
upto $i=n-1$

$$d_0\times b^0 + d_1\times b^1 + d_2\times b^2 + \ldots$$

$$327_8 = 3\times8^2 + 2\times8^1 + 7\times8^0$$
$$= 192 + 16 + 7$$
$$= \boxed{215_{10}}$$

How to convert from any base to base 10.

In base b, valid symbols are 0 to b-1.
If b>10, start using 'a', 'b', 'c', ...

b=16 is called hexadecimal
b=2 binary
b=8 octal
b=10 decimal

$$E3B_{16} = \boxed{14\times16^2 + 3\times16^1 + 11\times16^0}$$

Base 10 to Base b

$$215_{10} = \underline{327}_8$$

$$\begin{array}{r l}
8 \,\underline{|\, 215} & \\
8 \,\underline{|\, 26} & R\ 7 \\
8 \,\underline{|\, 3} & R\ 2 \\
0 & R\ 3
\end{array}$$

$$215 = \frac{d_2 \times 8^2 + d_1 \times 8^1 + d_0 \times 8^0}{8} \quad \cancel{\bmod 8}$$

$$215 \% 8 = d_0$$

$$215/8 = \boxed{d_2 \times 8^1 + d_1 \times 8^0}$$

↓↓↓

$327_8 = \longrightarrow$ 10 digits $\boxed{3|2|7}$

```
int val = 0;
for (i=0; i<numdigits; i++)
    val = 8*val + digits[i];
```

val ~~0~~
~~3~~
~~26~~
$\boxed{215}$

```
while (num > 0) {
    digit = num % 8;
    num /= 8;
}
```

1/26/17 ①

Base Conversion

① Other base → base 10

$326_7 = 3 \times 7^2 + 2 \times 7^1 + 6 \times 7^0$

$= 3 \times 49 + 14 + 6$

$= 147 + 14 + 6$

$= \boxed{167_{10}}$

② base 10 → other base

$$\begin{array}{r|l} 7 & 167 \\ 7 & 23 \quad R\ 6 \\ 7 & 3 \quad R\ 2 \\ & 0 \quad R\ 3 \end{array}$$

326_7

$\cancel{\le}\ 0 \qquad \%7$

$167 = (\boxed{\square \times 7^2 + \square \times 7^1} + \square \times 7^0)$

$167 \% 7 = d_0$

$d_2 \times 7^2 + d_1 \times 7^1 + \cancel{d_0 \times 7^0}$

$167 / 7 = \boxed{d_2 \times 7^1 + d_1 \times 7^0}$

③ base $b_1 \to b_2$

$b_1 \to 10 \to b_2$

```
int val = 0        strlen(num)
for (i=0; i<len; i++)
    val = base * val + value(num[i]);
```

327 $\qquad$ val ~~0~~

$\boxed{3}$

$3 \times 7 + 2 = \boxed{23}$

$23 \times 7 + 6 = 161 + 6$

$= \boxed{167}$

In CS, base 2, 8 + 16

binary, octal, hexadecimal

$AC7_{16} \to \boxed{}_2$

$\boxed{1010}\boxed{1100}\boxed{0111}_2$

Hex	Bin
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
A 10	1010
B 11	1011
C	1100
D	1101
E	1110
F	1111

$$2^{11}+2^{9}+2^{7}+2^{6}+2^{2}+2^{1}+2^{0}$$

$$= 2^{8}(2^{3}+2^{1})+2^{4}(2^{3}+2^{2})+2^{0}(2^{2}+2^{1}+2^{0})$$

$$= 16^{2}(A)+16^{1}(C)+16^{0}(7)$$

$AC7B_{16} \to \boxed{}_8$

$\boxed{1}\boxed{010}\boxed{110}\boxed{001}\boxed{111}\boxed{011}$

$1\ 2\ 6\ 1\ 7\ 3_8$

$b^{k} \to b \to b^{m}$

1/26/17 (3)

Given theoretical run times, we predicted actual run-times of algorithms.

Now, we need to learn <u>HOW</u> to do this!

Tool to help us find run-times: <u>Summations</u>

$$\underbrace{\sum_{i=a}^{b} f(i)}_{\text{Shorthand notation}} = f(a) + f(a+1) + f(a+2) + f(a+3) + \cdots + f(b).$$

```
int sum = 0, i;
for (i = a; i <= b; i++)
    sum += f(i);
```

$$\sum_{i=3}^{7} 2^i = 2^3 + 2^4 + 2^5 + 2^6 + 2^7$$

$$\sum_{i=1}^{5} (2i+1) = (2\cdot1+1) + (2\cdot2+1) + (2\cdot3+1) + (2\cdot4+1) + (2\cdot5+1)$$

$$= 3 + 5 + 7 + 9 + 11$$

$$\sum_{i=a}^{b} c = \underbrace{c + c + c + \cdots + c}_{\boxed{b-a+1}}$$

$$= (b-a+1)c$$

1/26/17 (4)

$$\sum_{i=a}^{b} c f(i) = c \cdot f(a) + c \cdot f(a+1) + c \cdot f(a+2) + \cdots + c f(b)$$
$$= c \left[f(a) + f(a+1) + \cdots + f(b) \right]$$
$$= c \sum_{i=a}^{b} f(i)$$

$$\sum_{i=a}^{b} (f(i) + g(i)) = f(a) + g(a) + f(a+1) + g(a+1) + \cdots f(b) + g(b)$$
$$= f(a) + f(a+1) + f(a+2) + \cdots + f(b) +$$
$$g(a) + g(a+1) + \cdots \quad g(b)$$
$$= \sum_{i=a}^{b} f(i) + \sum_{i=a}^{b} g(i)$$

$$\sum_{i=1}^{n} i = 1 + 2 + 3 + 4 + \cdots + n$$

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

$$(n+1) + (n+1) \; (n+1) + (n+1) \quad (n+1)$$

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

$$\frac{10100}{2} = \boxed{5050}$$

$$S = \sum_{i=0}^{\infty} x^i = 1 + x + x^2 + x^3 + \cdots$$

$|x| < 1$

$$-xS = \sum_{i=0}^{\infty} x^i \cdot x = \sum_{i=0}^{\infty} x^{i+1} = x + x^2 + x^3 + \cdots$$

$$S - xS = 1$$

$$S(1-x) = 1$$

$$\boxed{S = \frac{1}{1-x}}$$

Infinite geometric sequence

$$\sum_{i=0}^{\infty} 3 \cdot \left(\tfrac{1}{2}\right)^i = 3 \sum_{i=0}^{\infty} \left(\tfrac{1}{2}\right)^i$$

$$= 3 \cdot \frac{1}{1 - \frac{1}{2}}$$

$$= 3 \cdot \frac{1}{\frac{1}{2}}$$

$$= 3 \cdot 2 = \boxed{6}$$

$$\sum_{i=a}^{b} f(i) = \sum_{i=1}^{b} f(i) - \sum_{i=1}^{a-1} f(i)$$

$1 < a < b$

$$\boxed{f(1) + f(2) + \cdots + f(a-1) + f(a) + f(a+1) + \cdots + f(b)}$$

$$- \boxed{f(1) + f(2) + \cdots + f(a-1)}$$

$$f(a) + f(a+1) + \cdots + f(b)$$

$$\sum_{i=a}^{b} f(i)$$

Arithmetic Sequences – a sequence of #s with a common difference. 10, 13, 16, 19, ..

the sum of an arithmetic sequence of n terms $a_1, a_2, a_3, \dots, a_n$ = $\boxed{\dfrac{n(a_1+a_n)}{2}}$

If the common difference is d, then $a_n = a_1 + (n-1)d$

arith seq 1st term = 10
common diff = 3
terms = 20

$$a_{20} = a_1 + (20-1)3$$
$$= 10 + 19 \times 3$$
$$= 67$$

$$\text{Sum} = \frac{20(10+67)}{2} = 10 \times 77$$
$$= \boxed{770}$$

$$\sum_{i=10}^{3n+10} (2i+5) = \sum_{i=10}^{3n+10} 2i + \sum_{i=10}^{3n+10} 5$$

$$= 2\sum_{i=10}^{3n+10} i + 5(3n+10-10+1)$$

$$= 2\left[\sum_{i=1}^{3n+10} i - \sum_{i=1}^{9} i\right] + 5(3n+1)$$

$$= 2\cdot\frac{(3n+10)(3n+11)}{2} - 2(45) + 15n + 5$$

$$= 9n^2 + 30n + 33n + 110 - 90 + 15n + 5$$

$$= \boxed{9n^2 + 78n + 25}$$

$25 + 27 + 29 + \cdots$ 1/26/17 7

$a_1 = 25$, $a_{n'} = 2(3n+10)+5$

$= 6n+20+5$

$= \boxed{6n+25}$

$n' = 3n+10-10+1$

$= 3n+1$

$$\frac{(3n+1)}{2}(25+6n+25)$$

$$= \frac{(3n+1)}{2}(6n+50)$$

$$= (3n+1)(3n+25) = 9n^2+3n+75n+25$$

$$= \boxed{9n^2+78n+25}$$

✓

Analyzing Code Segments

```
for (i=0; i<n; i++) {
  O(1) do something simple
}
```

n times $O(n)$

$$\sum_{i=0}^{n-1} 1 = n$$

```
for (i=0; i<n; i++) {
  //O(1)
}
for (i=0; i<n; i++) {
  //O(1)
}
```

$O(n)$

$O(n)$

$O(n)$

```
for (i=0; i<n; i++) {
  for (j=0; j<n; j++) {
    //O(1)
  }
}
```

i	j	
0	0	
0	1	n
0	2	
0	3	
0	n-1	+
1	0	n
1	1	
1	2	+
1	n-1	n

$O(n^2)$

```
for (i=0; i<r; i++) {
  for (j=0; j<c; j++) {
    //O(1)
  }
}
```

$O(rc)$

```
for (i=0; i<n; i++) {
    for (j=0; j<=i; j++) {
        //O(1)
    }
}
```

$$\sum_{i=0}^{n-1}\left(\sum_{j=0}^{i} 1\right) = \sum_{i=0}^{n-1} (i+1)$$

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

$$O(n^2)$$