CSE 101-02 4-28-22

- Pa4: ext. 2 days (only)

Handout: Runtimes

Exercises at end:

~~(7)~~ (6) $f(n) + o(f(n)) = \Theta(f(n))$.

why? let $h(n) = o(f(n))$, so that

$\lim_{n\to\infty} \left(\frac{h(n)}{f(n)}\right) = 0$. Then

$$\lim_{n\to\infty} \left(\frac{f(n)+h(n)}{f(n)}\right) = \lim_{n\to\infty} \left(1 + \frac{h(n)}{f(n)}\right) = 1$$

$\therefore f(n) + h(n) = \Theta(f(n))$.

(7) compare 2^{3^n} to 3^{2^n}

$$\lim_{n\to\infty}\left(\frac{2^{3^n}}{3^{2^n}}\right) = \infty \quad \therefore \quad 2^{3^n} = \omega\left(3^{2^n}\right)$$

instead

$$\ln\left(\frac{2^{3^n}}{3^{2^n}}\right) = \ln\left(2^{3^n}\right) - \ln\left(3^{2^n}\right)$$

$$= 3^n \cdot \ln(2) - 2^n \cdot \ln(3)$$

$$= 3^n\left[\ln(2) - \frac{2^n}{3^n}\cdot\ln(3)\right]$$

$$\longrightarrow \infty$$

as $n\to\infty$

$$\therefore e^{\ln\left(\frac{2^{3^n}}{3^{2^n}}\right)} \longrightarrow e^{\infty} = \infty .$$

Ex
```
for i = 1 to n
    for j = 1 to i
        OP
```

$$(\#\ OPs) = 1 + 2 + 3 + \cdots + n = \frac{n(n+1)}{2}$$ ✓

let $S = 1 + 2 + 3 + \cdots + (n-2) + (n-1) + n$

$S = n + (n-1) + (n-2) + \cdots + 3 + 2 + 1$

$\therefore\ 2S = (n+1) + (n+1) + (n+1) + \cdots + (n+1) + (n+1) + (n+1)$

$\therefore\ 2S = n(n+1)$

$\therefore\ S = \frac{n(n+1)}{2}$.

so runtime $= \frac{1}{2}n^2 + \frac{1}{2}n = \Theta(n^2)$.

Ex SelectionSort(A)

```
1. n = length(A)
2. for i = 1 to (n-1)
3.     imin = i
4.     for j = i+1 to n
5.         if A[j] < A[imin]
6.             imin = j
7.     A[i] <-> A[imin]   // swap
```

$A_1 \ldots\ldots A_{i-1}$ | $A_i\ A_{i+1} \ldots\ldots A_n$

sorted | unsorted

Basic operation: comparison of array elements

$$(\#\text{comp}) = (n-1) + (n-2) + \cdots + 1$$

$$= 1 + 2 + \cdots + (n-2) + (n-1)$$

$$= \frac{(n-1)\left((n-1)+1\right)}{2}$$

$$= \frac{n(n-1)}{2}$$

$$= \frac{1}{2}n^2 - \frac{1}{2}n = \Theta(n^2).$$

Ex BFS: let $n = |V(G)|$, $m = |E(G)|$

cost of initialization $= \Theta(n)$

" " Queue ops $= \Theta(n)$

" " scanning adj lists $= \Theta(m)$

total cost $= \Theta(n+m)$

i.e. const. # bytes in adj list rep.

Ex DFS: let $n = |V(G)|$, $m = |E(G)|$.

initialization: $\Theta(n)$

main loop (other than Visit) = $\Theta(n)$

#calls to Visit = $\Theta(n)$

scanning adj lists = $\Theta(m)$

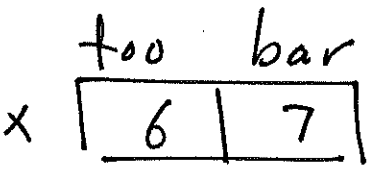
total cost = $\Theta(n+m)$

again linear in # bytes needed to store adj list rep.

Handout: ADTs in C++

A struct in C:

```
struct Blah{ // no typedef
    int foo;
    int bar;
};
:
struct Blah x;
x.foo = 6;
x.bar = 7
```



A struct in C++

```
struct Blah {
    int foo;
    int bar;
} Public by default

    Blah(int a, int b){
        foo = a;
        bar = b;
    }
};
:
Blah x(6,7);
```

x

foo	bar
6	7

A class in C++

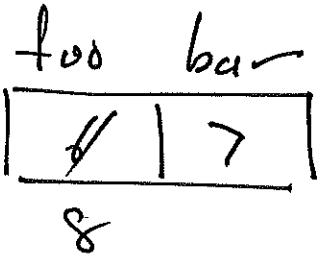
```
class Blah {
    int foo;
    int bar;
} private by default

    Blah(int a, int b){
        foo = a;
        bar = b;
    }

    void set_foo(int a){
        foo = a;
    }

};
```

```
:
:
Blah x(6,7);
x.set_foo(8);
```



basic skeleton for a C++ class:

```
class Blah{

private:
    // inner classes/structs
    // member variables
    // private helper fcns.

Public:
    // constructor(s), destructor
    // member methods
    // overloaded operators
};
```