Midterm review (brief)

C++:

- "big three"
- operators

A3 = A1 + A2;

A3.operator=(A1.operator+(A2))

- iterators

- big-oh:
  - know formal def's of
    - $O(T(n))$
    - $\Theta(T(n))$
    - $\Omega(T(n))$
  - know meaning
  - practical implications
    (e.g. compare: contrast algs)

the algs: (complete, balanced BST)

how many nodes per level?

(per height)

○ $n = 1$

~~$2^0, h = 0$~~

$h = 1, \quad 2^1 - 1 = 1$

?

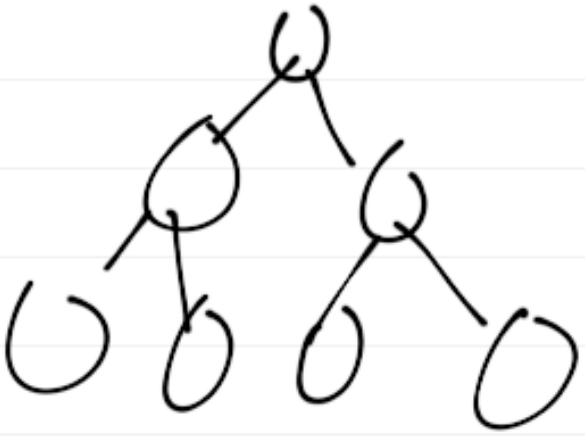
$n = 3$

~~$h = 1$~~

~~$2^1 = 2 + 1$~~

$h = 2$

$2^2 - 1 = 3$

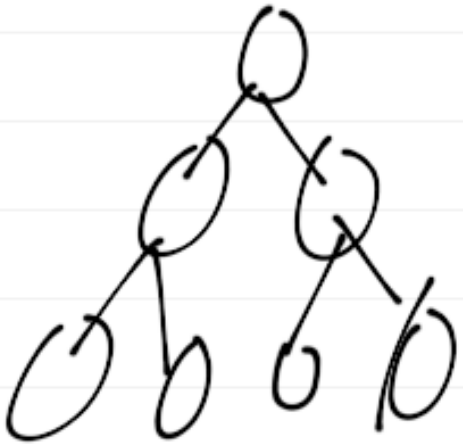


$n = 7$

$h = 3$

$2^3 - 1 = 7$

tree algs: why $\Omega(\lg n)$?



n nodes,

$2^h - 1$ nodes

in tree w/

height h,

search for item:

only need to go as

far as h, & $(2^h - 1) \div \lg(n)$

- binary trees
- know defs, properties
- what does it mean for a binary tree to be complete?

- heaps & heapsort

- how fast is it?

(O, Θ, Ω)

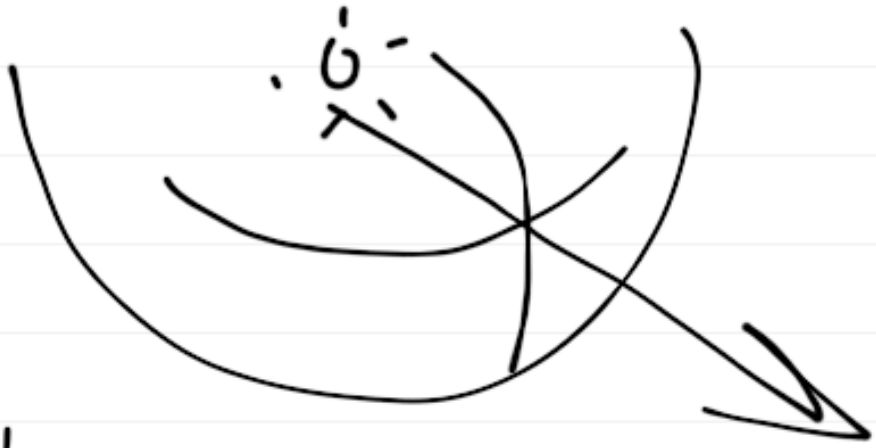
for n items

AVL trees

- know how to insert, delete, do rotations ("on paper", not code)

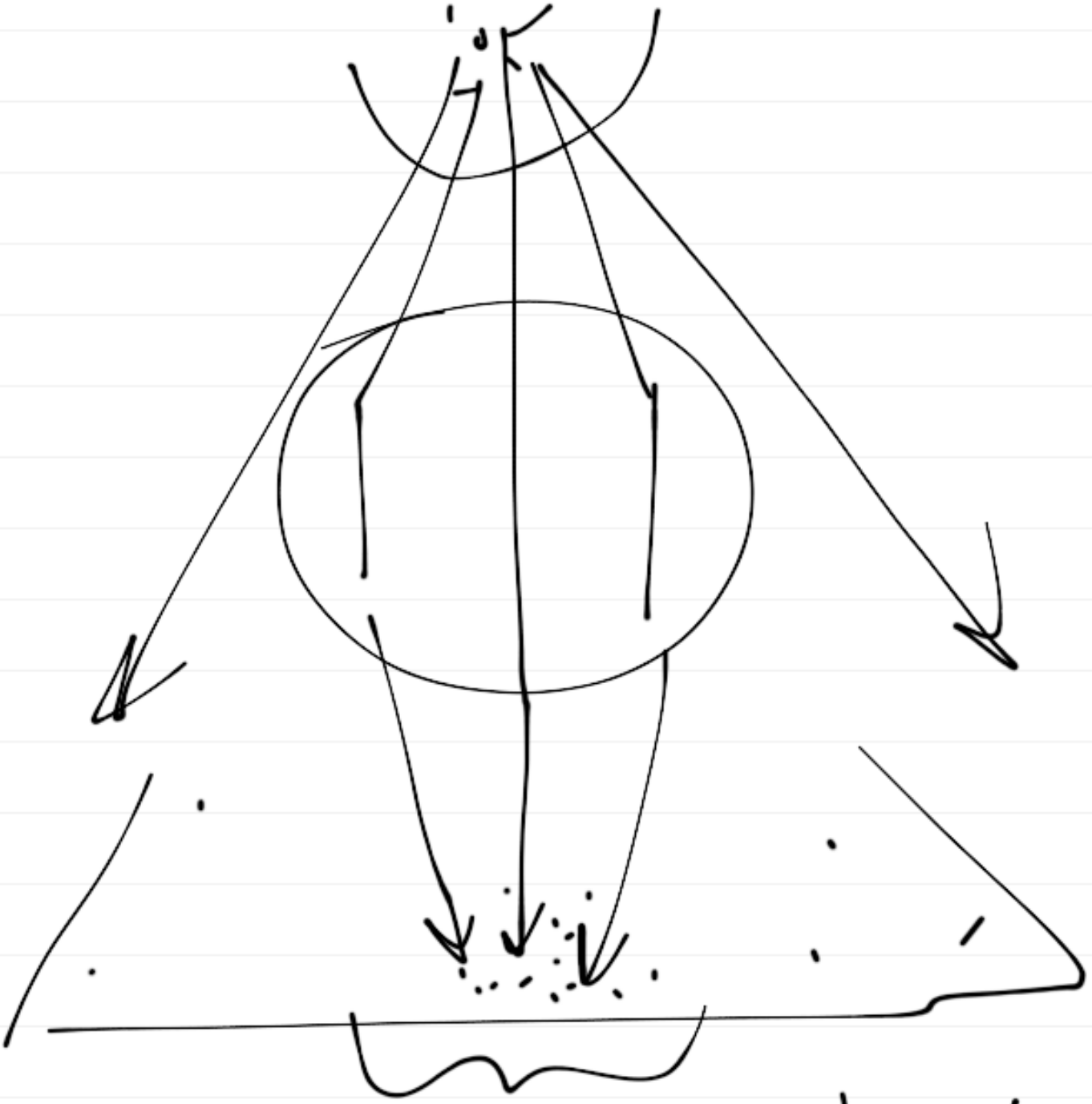
ASG 3: AVL tree

ASG 4: photon emission



"Monte Carlo simulation"

random dir., azimuth



denser concentration of photons ≡ caustic

Alg:

1. shoot photons
   (in random dir.)
   (from light source)

2. ray trace

photons:

- can "stick" to surfaces
- can reflect
- can transmit

- photons are like rays (derive photon scatters from parent ray)

- in ASCII: just "shoot" photons, and those that stick get added to std::vector (array)

- in ASG4: list of print out photos

Related labs:

write an OpenGL viewer to "see" (render) photos! (learn Qt, learn OpenGL)

- event cells,

ASG 6:

insert photons into

kd-tree, ray trace

↓

knn-query