Training II

2021. Sept 19. Ye Cao.

Review of C++ pointer.

& — Used to get the memory location of a variable

* — Used to get the value of the variable associated the pointer

Question:

```
int x = 10;
int *P = &x;
cout << p << endl;
```

1. Is printout equal 10?
2. If not, what's fix to print 10?

Review of pair

```
pair<int, int> p;   // represents a point
p.first;            // x-coordinate
p.second;           // y-coordinate
```

Review of sort

<7 5 6 4 10 1 3>

<1 3 4 5 6 7 10>

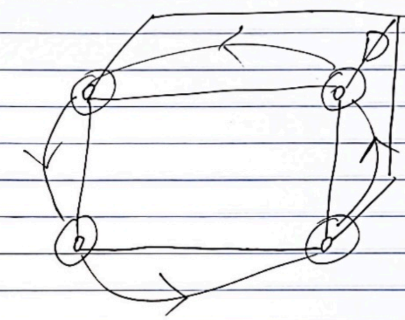
Review of binary search:

1. Define active region
2. Execute search
3. Update active region

Aquarium Tank Problem :

Problem Understanding :

Inputs : 1. Cartegian Coordinate of the convex polygon in counterclockwise order.



2. Depth D

3. Water volume L.

Computational Procedure :

1. How to process inputs ?

put into pairs contained by an array.

2. How to calculate height of water ?

GUESSING vs Math Formula

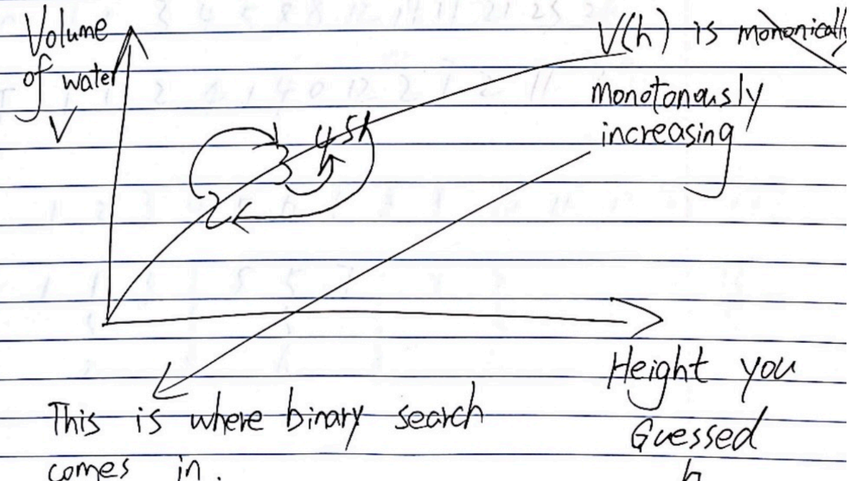
linear guess – $O(N)$ $O(1)$

Binary Search – $O(\log(N))$

3. Look at input constraints. Gussing is Feasible

Tips:

1. Guessing is a very generalized approach in CP. Normally you don't know the answer, you make a guess and see if you can make a more EDUCATED guess each time.
2. Generally, guessing is very good if each time you make a guess, you are able to eliminate a range of answers from being right ones besides the single guess you made. This is the core idea of binary search.

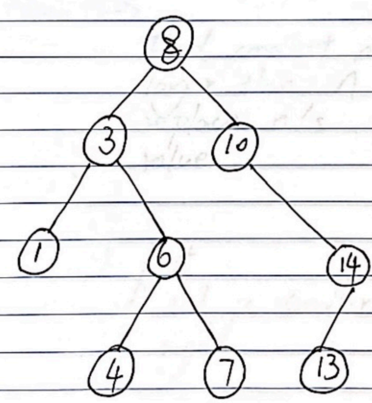


Intro to a tree

In computer science, a tree is an abstract data structure that simulates a hierarchical tree structure, with a root and subtree of children with parent node, represented as a set of linked nodes.

In graph theory, a tree is a connected acyclic graph.

Binary Search Tree



Insertion

Deletion

Retrieval

Insertion

1. Start with root
2. Go left if val is smaller
3. Otherwise go right
4. If reach a leaf node, attach val to leaf node

Deletion .

No children → Remove node n

One children → Remove node n and connect n's parent to n's children

Two children → Find smallest node m that's larger than n, remove m and replace n's value with m's value.

Retrieval O(h)

1. Start with root
2. Go left if val is smaller
3. Otherwise go right
4. If reach a leaf node, not found .

Binary Indexed Tree / Fenwick Tree

Proposed by Peter M. Fenwick

Range Sum Query

Point Update

Consider this problem: There're n boxes and 2 ques

1. add marble to box i
2. sum marbles from box k to box r.

How to come up with a data structure that works in time complexity $O(m \cdot \log(n))$?

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
f	1	0	2	1	1	3	0	4	2	5	2	2	3	1	0	2
c	1	1	3	4	5	8	8	12	14	19	21	23	26	27	27	29
T	1	1	2	4	1	4	0	12	2	7	2	11	3	4	0	29

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	3	1	5	5	7	1	9	9	11	9	13	13	15
	~		~		~		~		~		~	~	~	
	2		4		6		8		10		12	·	14	

Binary Index Tree Implementation Details

1. Isolate last bit

$$\text{LastBit}(10010) = 10$$
$$(11000) = 1000$$
$$(10001) = 1$$
$$(10000) = 10000$$

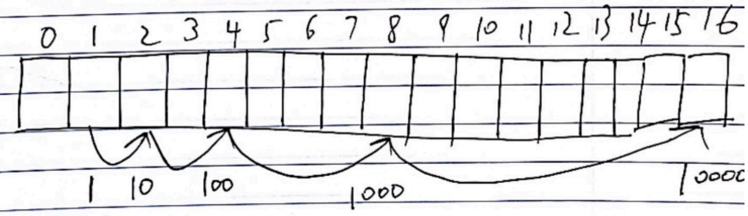
Last bit
$a1b$
Before last bit
zeros

$$(a1b)^{-1} = a^{-}0b^{-} + 1$$
$$= a^{-1}0(111\cdots1) + 1$$
$$= a^{-}1b$$

$$\therefore\ a1b \,\&\, (a1b)^{-} = \text{Last bit}$$

2. Update

Start with current idx + 1 (1-based index)
Continue update to parent index




```
idx ++;    while(idx <= n)
               idx += idx & (-idx);
```

3. Getsum

Start with current idx + 1 (1-based index)
Continue update to ~~parent~~ child index