Training I

2021. Sept 11. Ye. Cao.

1. Concept Review C++

C++ variables Type Pointer
    int , char , float , double , bool , long long

Logic Flow
    if else   &&, ||

Loop
    while , for

I/O
    scanf , cin , printf , cout , getline(cin, s)

Array
    int arr[10] ;   vector <int> arr_vector [10] ;
    int arr[100][10];   vector <int> edges [10] ;

    Creates a 100x10 array

Pointer example   (Pointer arithmetics)

```
int var[3] = {10, 100, 200};
int * ptr = var;                    → Why not & before var?

for (int i=0; i < 3; i++){          ←— Because array
    cout << *ptr << endl;              itself is a mem
    ptr ++;                            address, and when
}                                      you do arr[0] you
                                       dereference it.
```

$p[6] = \{1, 2, 3, 4, 5, 6\}$

$int \; *q = (int \; *)(p+1);$

$*q == 2$

$p+x \iff (int \; *)((*char)p +$

$sizeof(*p) * x)$

Cast to pointer This is a pointer to second slot of
type, but it's
not necessary array P.
here, because
(p+1) is a
pointer itself, here it's just specifying point type is int.

All rigtht cool, but why should I give a damn?

Struct:

```
struct Node {
        int data;
        node * next;
};
```

Node	
int	node*
data	next

Mergesort

1. Split

| 2 | 3 | 1 | 4 | 5 | 6 | 7 | 8 |

| | | | | | | | |

| 2 | 3 | 1 | 4 | 6 | 7 | 5 | 8 |

2. Merge

| 2 | 3 | 1 | 4 | 6 | 7 | 5 | 8 |

| 2 | 3 | 1 | 4 | 6 | 7 | 5 | 8 |

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |

(Normally, you wouldn't just want to sort)

Search

lower_bound (begin , end)
upper_bound (begin , end)

1. assume arr is already sorted

2. return an iterator (pointer) to the array

3. you have to use pointer arithmetic

4. Dereference * to see the pointer's content

Generalized binary search

An important use for binary search is to find the position
where the value of a function changes!

x	0	1	.....	$K-1$	R	$k+1$	....
$6R(x)$	F	F		F	T	T	

```
int x = -1;
for ( int b = 2, b ≥ 1 ; b /= 2 ) {
    while ( ! oR (x+b) ) x += b;
}
int R = x + 1;
```

Tricks for sort

1. Sort pairs is more often than sorting single values
   tuples

2. With sorting tuples, you can do LOTS of things.

Generalized binary search

Usual way : check middle element of active region
            if target
                    terminates
            else
                    update active region

Alternative way :
                 Make jumps and slow the speed
                 when getting closer to target
                 element.

EX. Finding maximum value :

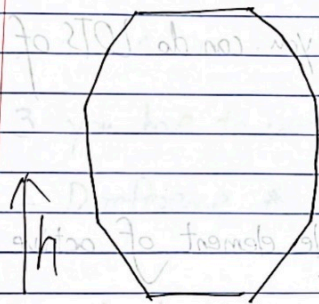$$f(x) < f(x+1) \text{ for } x < R \text{ and}$$
$$f(x) > f(x+1) \text{ for } x \geq R$$

```
int x = -1 ;
for ( int b = 8, b >= 1 ; b /= 2 )                    R = x+1
    while (f(x+b) < f(x+b+1) x += b ;
```
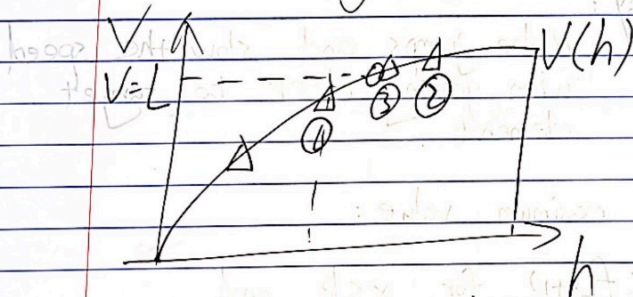
Aquarium Example

Main Takeaway :

Generalize bisearch techniques in other senarios

In this case the volume of a gemetric object grows monotonously, thus we can use bisearch to find the CHANGE POINT
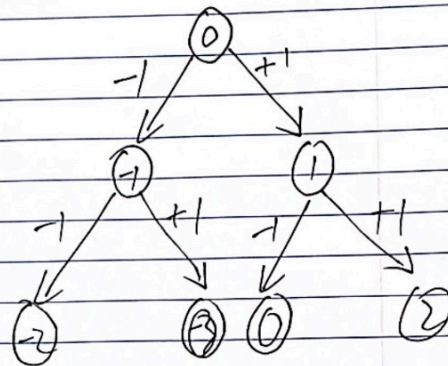

$V = L$ ... $V(h)$

h

Inverse Problem: $V^{-1}(L)$ Find point h_1 such that $V(h_1) = L$

Leetcode Problem

1. BFS + Sorting

    More left
    More up

2. Tuple (leftness,
            upness,
            val)

3. Sort (Tuple List)

4. Print

Root is 0, -1 go left
                +1 go right
Using BFS to make sure that
nodes are marked with magnitude
from left to right.