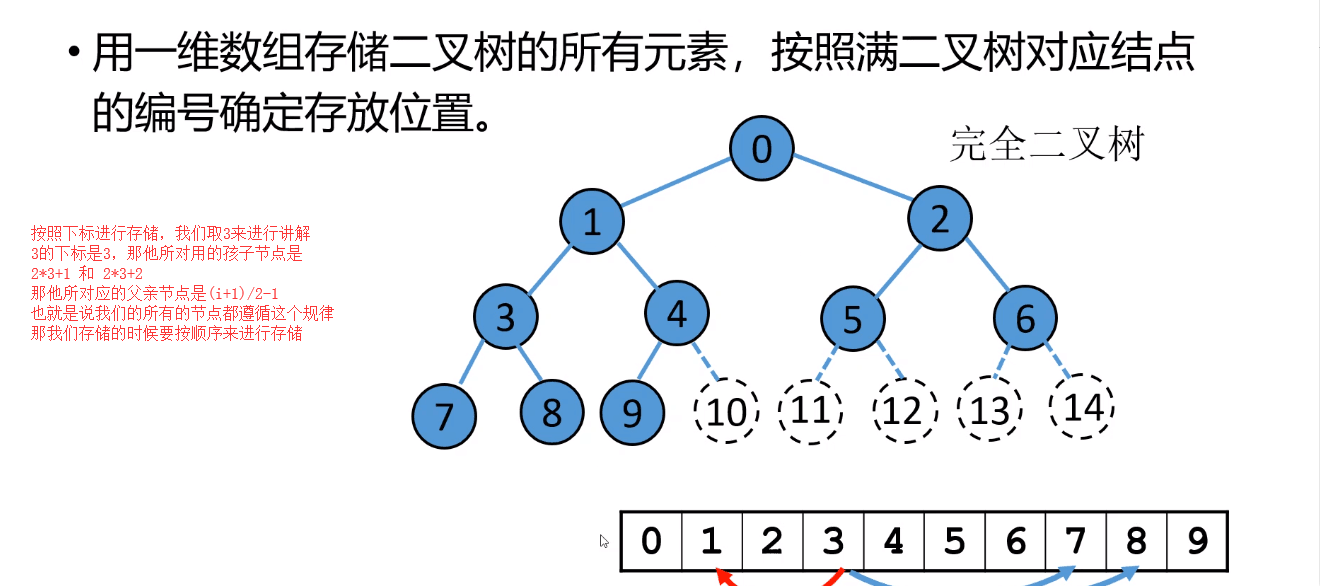
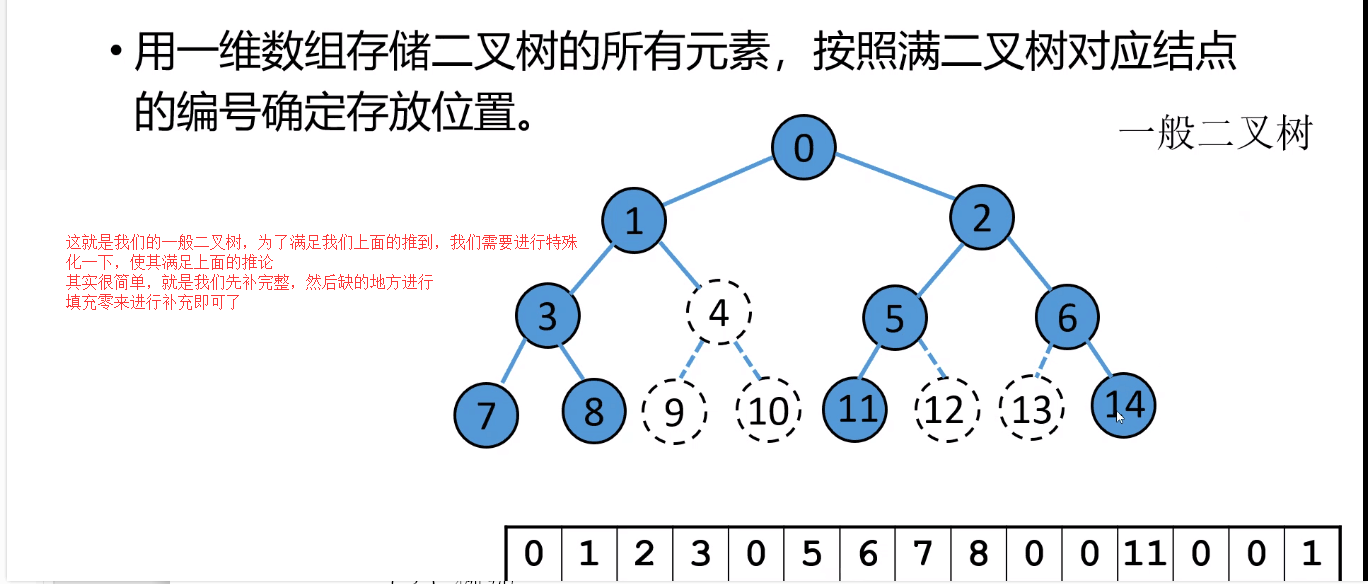
**二叉树的顺序存储**

* 1. 二叉树
     1. 图解



* + 1. 不过，它也有缺陷，如果二叉树不是满二叉树或者完全二叉树，那么空间的利用率非常低
    2. 例如



* + 1. 代码实现

#pragma once

#include <iostream>

#include <vector>

using namespace std;

template <typename T>

class Tree {

int size;

vector<T> data;

public:

Tree(int cap):size(cap) {

if (size) {

data.resize(size);

}

}

// 不需要修改元数据，所以用const放在末尾进行修饰

T& operator[](int i)const {

return data[i];

}

T& operator[](int i) {

return data[i];

}

int left\_child(int i) { return 2 \* i + 1; }

int right\_child(int i) { return 2 \* i + 2; }

int current\_parent(int i) { return (i+1)/2-1; }

//遍历放在后面进行讲解

void traverse() {

//TODO

}

};