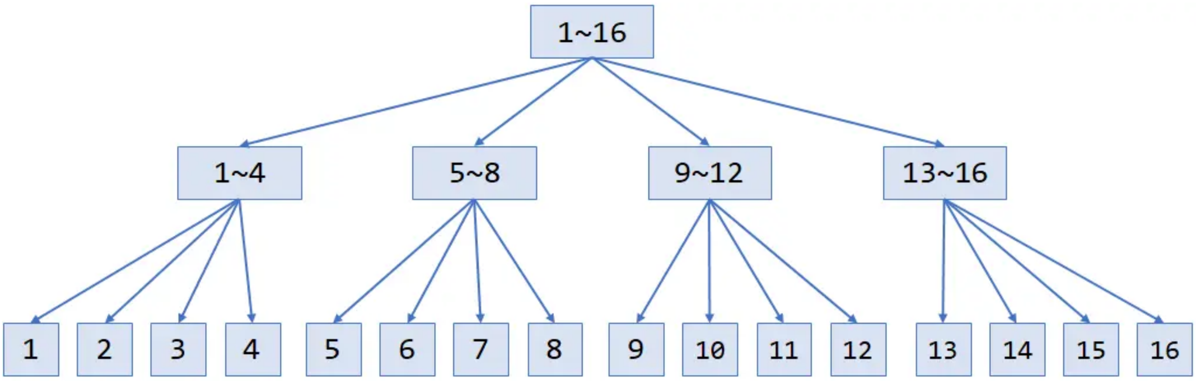
分块

与其说分块是一种算法，不如说**分块**是一种思想，把一个整体划分为若干个小块，对整块整体处理，零散块单独处理。我们将着重于**块状数组**的介绍

块状数组把一个长度为 b 的数组划分为 a 块，每块长度为 $\frac{n}{a}$。对于一次区间操作，对区间内部的整块进行整体的操作，对区间边缘的零散块单独暴力处理。（所以分块被称为"优雅的暴力"）

这里，块数既不能太少也不能太多。如果太少，区间中整块的数量会很少，我们要花费大量时间处理零散块；如果太多，又会让块的长度太短，失去整体处理的意义。一般来说，我们取块数为 $\sqrt{n}$，这样在最坏情况下，我们要处理接近 $\sqrt{n}$ 个整块，还要对长度为 $\frac{2n}{\sqrt{n}} = 2\sqrt{n}$ 的零散块单独处理，总时间复杂度为 $O(\sqrt{n})$。这是一种**根号算法。**

显然，分块的时间复杂度比不上线段树或树状数组这些**对数级算法**。但由此换来的，是更高的灵活性。与线段树不同，块状数组并不要求所维护信息满足结合律，也不需要一层层地传递标记。但它们又有相似之处，线段树是一棵高度约为 $\log n$ 的树，而块状数组则可被看成一棵高度为 3 的树：



和所有的其它数据结构一样，我们的分块内的具体操作也是需要具体问题具体分析的

但分块的本质是：**完整块完整处理，不完整块单独处理**

代码实现

一开始，我们分块的块大小规定如下（注意数组是 $0-\text{index}$）：

```
// n: 数组大小
int n; cin >> n;
// sq: 块大小
int sq = sqrt(n) + 1; cmin(sq, n);
```

对于某个位置 i，他所在的**块编号**就是 $\lfloor \frac{i}{sq} \rfloor$

```
auto idx = i / sq;
```

对于某个**区间操作**：

```
// 左边边角
while (l <= r && l % sq) {
    a[l] xxx    // do something
    l++;
}
// 右边边角
```

```
while (r >= l && r % sq != sq - 1) {
    a[r] xxx    // do something
    r--;
}

// 整块
l = l / sq, r = (r + 1) / sq;
for (int i = l; i < r; i++) {
    tag[i] xxx // do something
}
```