1、placement new

<https://zhuanlan.zhihu.com/p/359503982>

创建对象(调用该类的构造函数)但是不分配内存，而是在已有的内存块上面创建对象。用于需要反复创建并删除的对象上，可以降低分配释放内存的性能消耗。

2、C++中虚函数可以是内联函数吗？

<https://www.jianshu.com/p/84a8335444dd>

inline virtual 唯一可以内联的时候是：编译的时候知道所调用的对象是哪个类。

3、自由存储区和堆

<https://blog.csdn.net/Cookey_July/article/details/81209352>

堆是操作系统维护的一块内存，是一个物理概念，而自由存储是C++中通过new与delete动态分配和释放的对象的存储区，是一个逻辑概念。

4、\_\_thread

<https://www.jianshu.com/p/997b533842c8>

\_\_thread是GCC内置的线程局部存储设施。\_thread变量每一个线程有一份独立实体，各个线程的值互不干扰。可以用来修饰那些带有全局性且值可能变，但是又不值得用全局变量保护的变量。

5、extern “C”

<https://www.bbsmax.com/A/QW5YQY335m/>

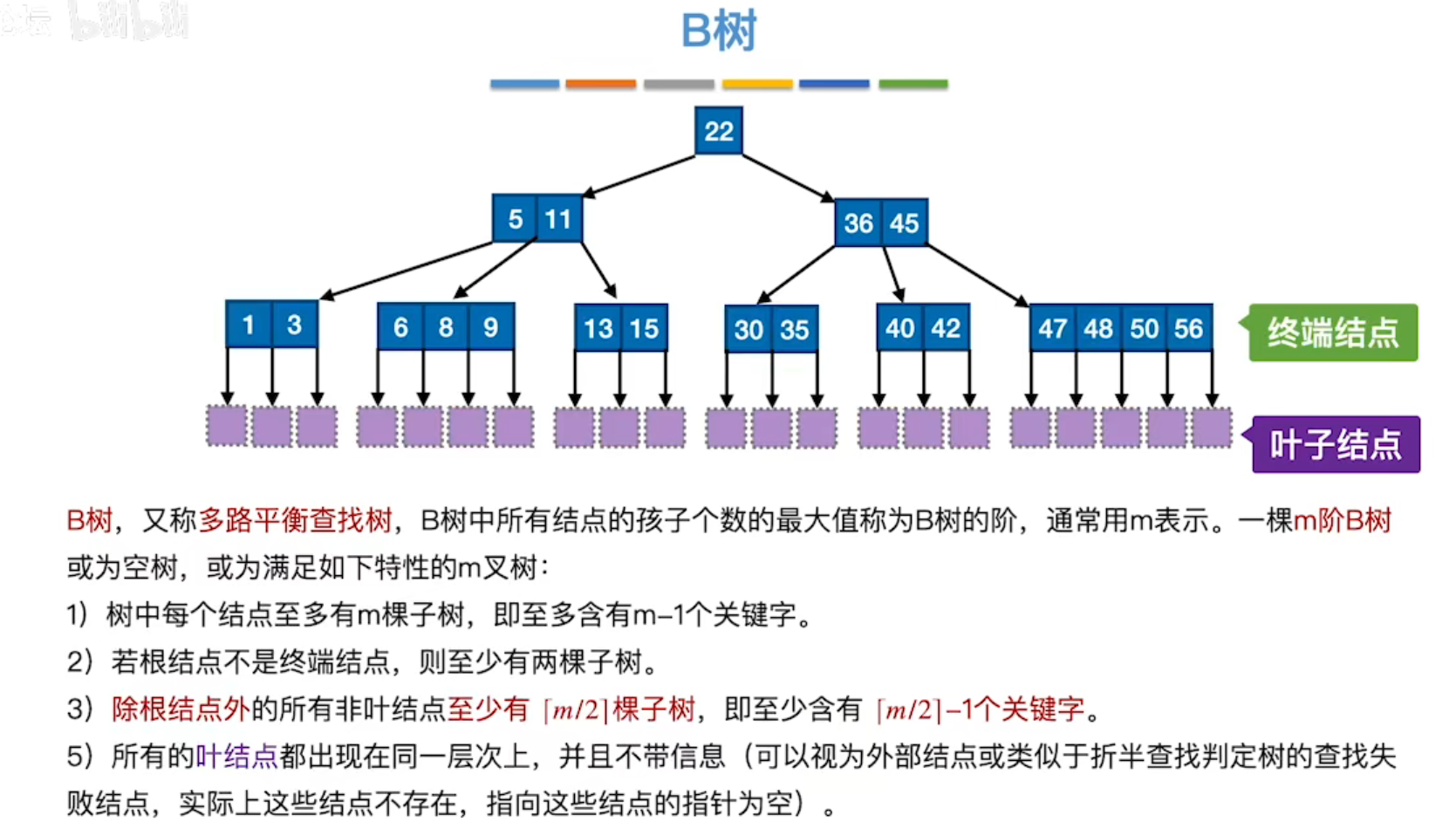
<https://www.cnblogs.com/skynet/archive/2010/07/10/1774964.html>

一句话：用于C和C++语言混合编程，因为C语言编译器和C++语言编译器在目标文件符号表中对函数名称的处理方式不同，如果C++语言直接调用C，会找不到符号而发生链接

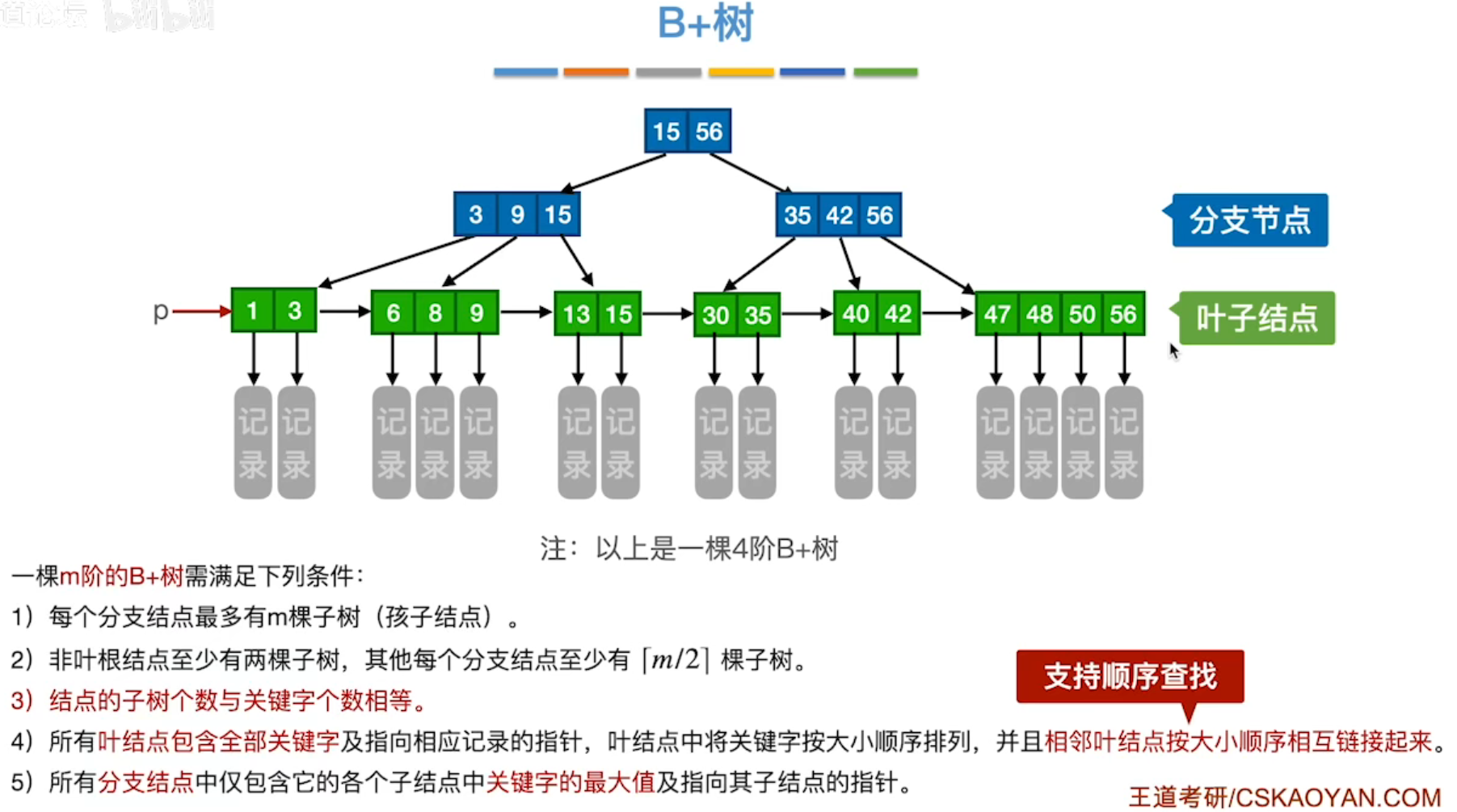
错误。在C++源文件中的语句前面加上extern "C"，表明它按照类C的编译和连接规约来编译和连接，而不是C++的编译的连接规约。这样在类C的代码中就可以调用C++的函数or变量等。

6、B树、B+树

多路平衡查找树：



7、B+树



B树与B+树的区别：

（1）查找：B树不需要，根据关键字可能在中间某一层的某个节点就找到了记录地址。B+树最终一定要找到叶子节点那一层，才能找到记录地址，进而读取记录。

（2）关键字个数：

B树根节点关键字个数：[1, m - 1]，其他节点关键字个数[m/2- 1, m - 1]

B树根节点关键字个数：[1, m]，其他节点关键字个数[m/2, m]

（3）B树中的每个关键字只出现一次，B+树中关键字可能出现多次。

在B+树中，非叶结点不含有该关键字对应记录的存储地址。可以使一个磁盘块可以包含更多个关键字，使得B+树的阶更大，树高更矮，读磁盘次数更少，查找更快。



8、红黑树

协程