在线性搜索中，系统扫描每个文件块并测试所有记录，以确定它们是否满足选择条件。要访问文件的第一个块，需要进行初始搜索。如果文件块不是连续存储的，可能需要额外的查找，但是为了简单起见，我们忽略了这个影响。

虽然它可能比其他实现选择的算法要慢，但线性搜索算法可以应用于任何文件，不管文件的顺序、索引的可用性或选择操作的性质如何。我们将要学习的其他算法并不适用于所有的情况，但是当适用时，它们通常比线性搜索要快。

线性扫描以及其他选择算法的成本估计如图12.3所示。在图中，我们使用hi来表示B+-树的高度。现实生活中的优化器通常假设树的根位于内存缓冲区中，因为它经常被访问。一些优化器甚至假设，除了叶节点之外，树的所有叶节点都存在于内存中，因为它们被访问的频率相对较高，而且通常只有不到1%的B+-树节点是非叶节点。成本公式可以适当修改。

索引结构被称为访问路径，因为它们提供了一个可以定位和访问数据的路径。在第11章中，我们指出按与物理顺序紧密对应的顺序读取文件的记录是有效的。还记得主索引(也称为集群索引)是一个允许按与文件中的物理顺序相对应的顺序读取文件记录的索引。不是主索引的索引称为辅助索引。

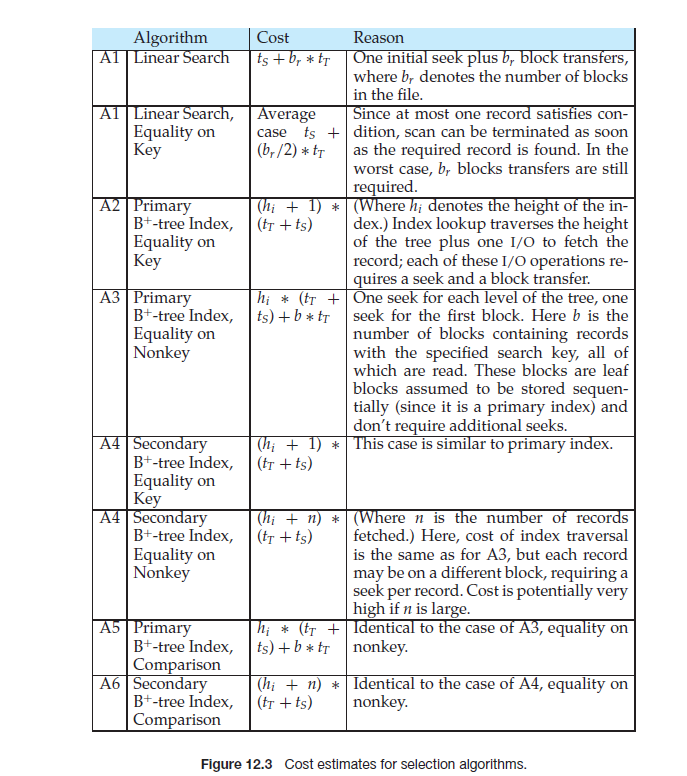
使用索引的搜索算法称为索引扫描。我们使用选择谓词来指导我们选择用于处理查询的索引。使用索引的搜索算法有:

* A2(主索引，键上相等)。对于键属性与主索引的相等性比较，我们可以使用索引检索满足相应相等条件的单个记录。费用概算见图12.3。
* A3(主索引，非键上相等)。当选择条件指定了非键属性a上的相等比较时，我们可以通过主索引检索多条记录。但是，记录必须连续存储在文件中，因为文件是根据搜索键排序的。费用概算见图12.3。
* A4(二级索引，相等)。指定相等条件的选择可以使用二级索引。如果键上有相等条件，此策略可以检索单个记录;如果索引字段不是键，则可以检索多个记录。

在第一种情况下，只检索一条记录。这种情况下的时间成本与主索引(情况A2)相同。

在第二种情况下，每个记录可能驻留在不同的块上，这可能导致每个检索到的记录都有一个I/O操作，每个I/O操作都需要一个查找和一个块传输。在这种情况下，最坏情况的时间成本是(hi + n)∗(tS + tT)，其中n是获取的记录的数量，如果每个记录位于不同的磁盘块中，并且块的获取是随机排序的。如果检索到大量的记录，最坏情况下的代价可能比线性搜索更糟。

如果内存缓冲区很大，则包含记录的块可能已经在缓冲区中了。通过考虑包含已在缓冲区中的记录的块的概率，可以构建对选择的平均或预期成本的估计。对于较大的缓冲区，这个估计值将远远小于最坏情况下的估计值。



hi B+树高度

tT 磁盘子系统传输数据块平均时间

tS 平均块访问时间(磁盘寻道时间加上旋转延迟)

b 数据块数量

S 执行的搜索次数

在某些算法中，包括A2，使用B+树文件组织可以保存一次访问，因为记录存储在树的叶级。

如第11.4.2节所述，当记录存储在B+-tree文件组织或其他可能需要重定位记录的文件组织中时，辅助索引通常不存储指向这些记录的指针。相反，二级索引存储B+树文件组织中用作搜索键的属性值。通过这样的辅助索引访问记录的开销会更大:首先搜索辅助索引来查找主索引的搜索键值，然后查找主索引来查找记录。如果要使用次级指数，就必须适当修改所述的费用公式。