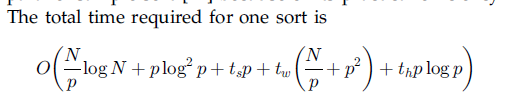
KD tree问题：

1. 边界问题：

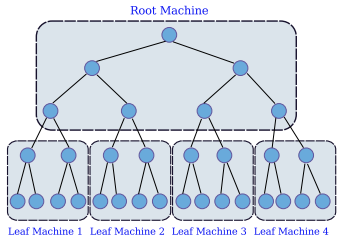
如果点 是小于等于的，刚好在边界，这时候如何找它的左下方点。

2． TPDS有一些 分median的算法。Parallel Tree Construction for log p Levels 还有并行sort然后median的算法。Mesh和hypercube是什么东西？

3. node之间通信问题，如何decent的表示。

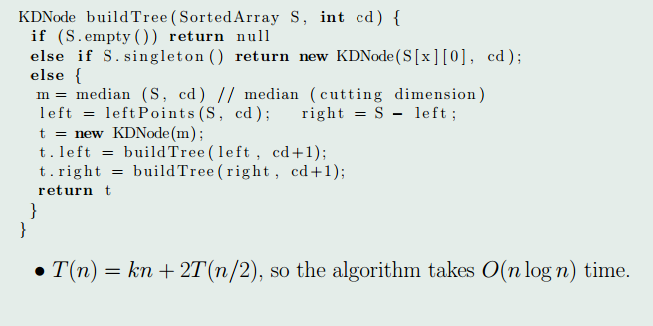
4. 从哪个区域开始扫描 可以最优化。

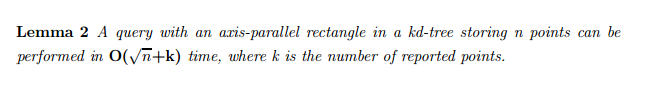
5. Rtree 是可以交叠的么？

6. 

Building a static *k*-d tree from *n* points takes:

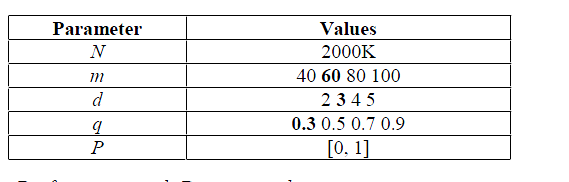
* [O](http://en.wikipedia.org/wiki/Big_O_notation)(*n* log2 *n*) time if an O(*n* log *n*) sort such as [Heapsort](http://en.wikipedia.org/wiki/Heapsort" \o "Heapsort) is used to compute the median at each level;
* O(*n* log *n*) time if a complex linear-time [median-finding](http://en.wikipedia.org/wiki/Selection_algorithm) algorithm such as the one described in Cormen *et al.*[[7]](http://en.wikipedia.org/wiki/K-d_tree#cite_note-7) is used;
* O(*kn* log *n*) plus O([*k*-1]*n* log *n*) time if *n* points are sorted in each of *k* dimensions using an O(*n* log *n*) sort *prior* to building the *k*-d tree.





7. parallel那篇一直在讲的是communication cost，因为建立K-Dtree，要不停的sort，swap points。所以communication cost还是挺大的。可以只写一个复杂度，然后不去和任何文章比较。

一些实验比较：1. Z-tree，最多D==8-dimensional，N=20K

1. PODS， dimension=[1,1000], N =20K
2. ICDCS
3. 可以强调structured P2P networks， for web information systems.

Each Server can store any subset of DB.

想到一个可以优化的地方，可以先找到某个 object的skyline，然后其他 点就是它的重合地方。类似于 Jian Pei 里面的idea。给skyline分层，然后一层一层的求, overhead会变小

还有一种优化的办法，用已经算好的object的delta，去求他dominate其他object的概率，还没想好怎么算。

ICDCS他们的数据是distributed，

In fact, It is inconvenient for Probabilistic Skyline working upon Hadoop Framework:

1. Data is needed to be operated duplicately, since one instance is needed to be compared with all instances of all objects.
2. It is hard to prune in Hadoop environment. Therefore, the efficiency might be low in MapReduce.
3. Process must always stand in one machine: index should be held all time. This property is suitable for distributed circumstances.