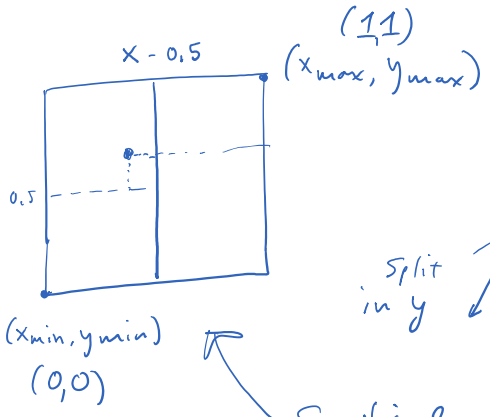
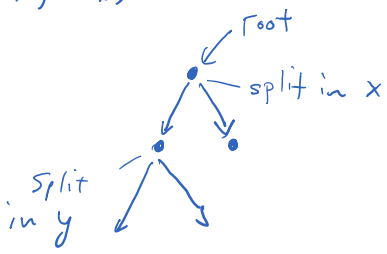
Tree structures...

Monday, September 25, 2017 1:09 PM

Partition



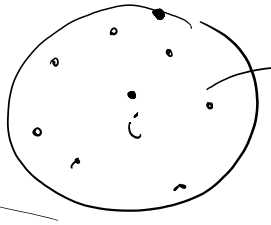


Spatial Binary Tree
every 3rd level ≡ octtree (in 3D)

If you use the median of the {x, y, z} values, then you get a k-D tree.

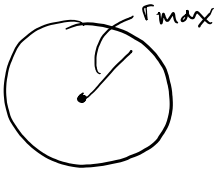
very fast for searching

Nearest Neighbor Searching

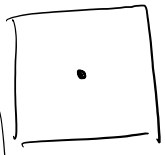


find the k-closest particles

~2X harder easier



find all particles within rmax.

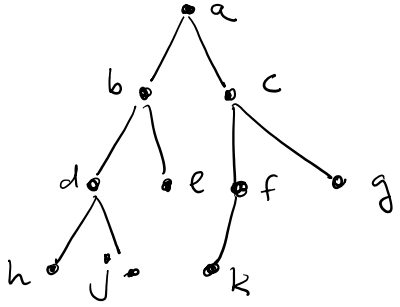


another type of query

Recursion

LNR

```
lnr(p) {
  if (p != NULL);
    lnr(p->left);
    print(p->letter);
  }lnr(p->right);
```

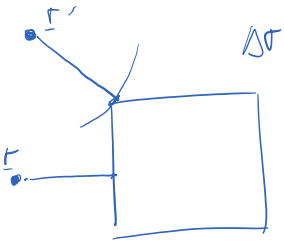


hdjbeakfcg

NLR: abdhjecfkg

hdjbeaktcg

```
NN(p) {
    if ( dist2(r, p->left) < dist2(r, p->right) ) {
        NN(p->left);
        NN(p->right);
    }
    else {
        NN(p->right);
        NN(p->left);
    }
}
```



$\Delta r^2 = \Delta x^2 + \Delta y^2$

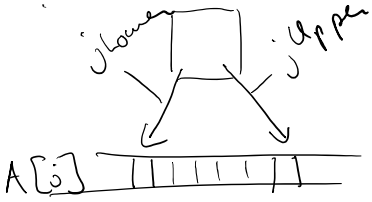
Sometimes 0 Sometimes 0

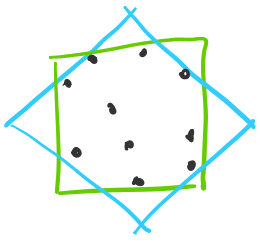
```
class bnd {
    float xmin[2];
    float xmax[2];
}
```

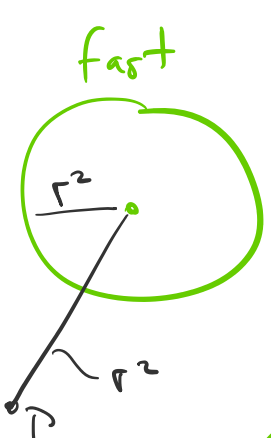
dimension

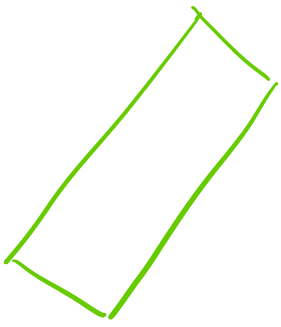
```
int BALLWALK(p, r, r2max) {
    int cnt = 0;
    if (isLeaf(p)) {
        for (j = p->jLower; j <= p->jUpper; ++j)
            if (dist2(r, A[j].r) < r2max)
                ++cnt;
    }
    else {
        if (dist2(r, p->left) < r2max) {
            cnt += BALLWALK(p->left, r, r2max);
        }
        if (dist2(r, p->right) < r2max) {
            cnt += BALLWALK(p->right, r, r2max);
        }
    }
```

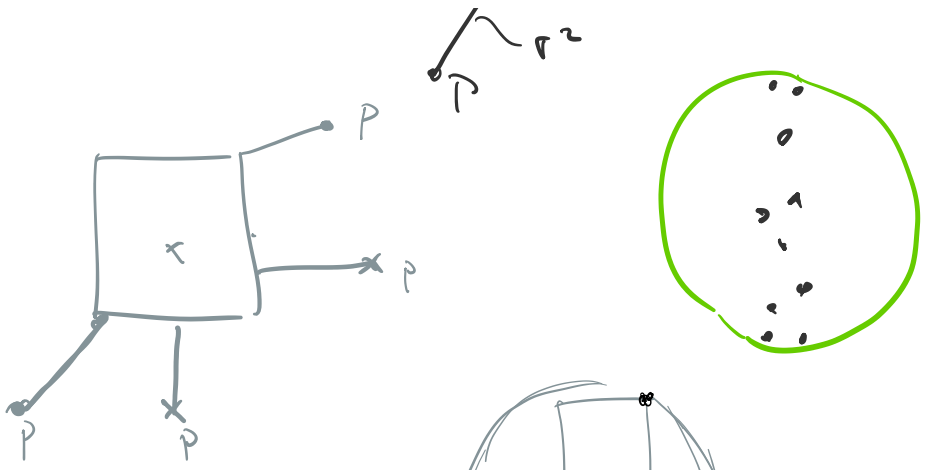
query

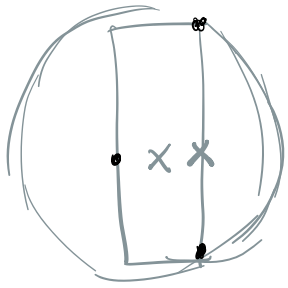


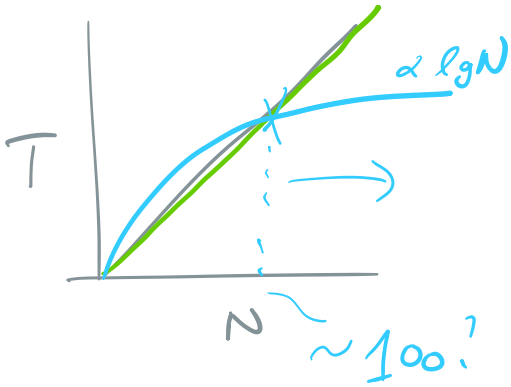






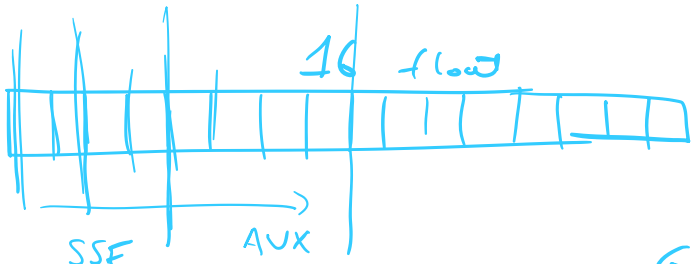






float 32

AUX 512



* +

GPU

32 float SM

MUL FMA FMA

$r2 = x_i * x_i + y_i * y_i + z_i * z_i;$

AoS

SoA

AoSoA