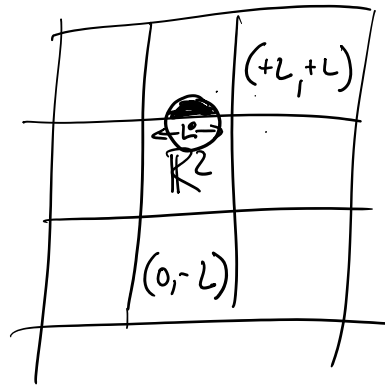
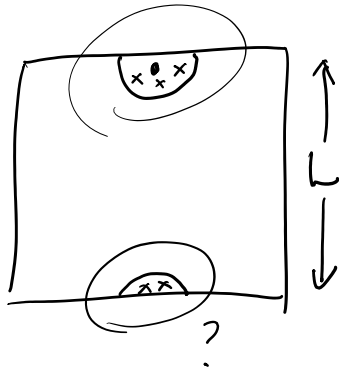


Periodic BC's



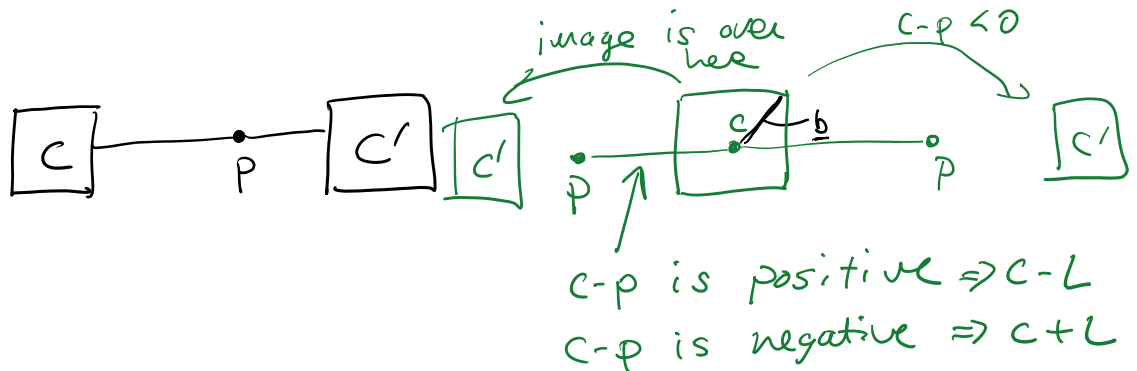
offset
vector

Ballwalk (....., offset)

↑ gets an extra parameter

★ Call 9 times

OR



$d2 = 0$

for ($d=0; d<2; ++d$) {

$t = c[d] - p[d]$

if ($t < 0$) $t1 = t + L$

else $t1 = t - L$

$t = \text{abs}(t) - b[d]$

$t1 = \text{abs}(t1) - b[d]$

if ($t1 < t$) $t = t1$

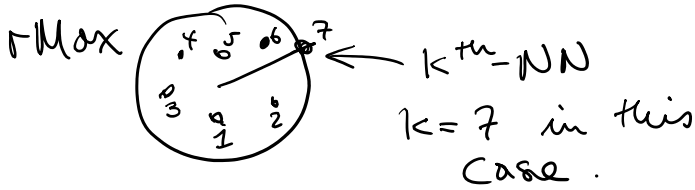
if ($t > 0$) $d2 += t * t$

}

return $d2$

L nearest neighbours (kNN)

Finding k-Nearest Neighbors (kNN)

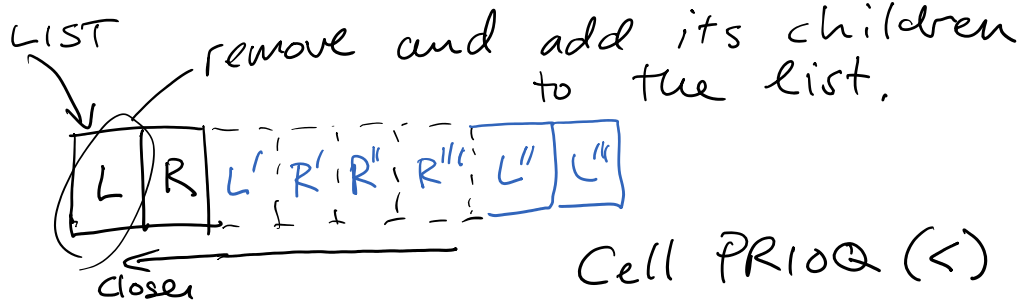
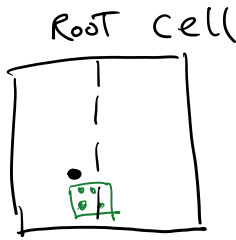


Recursion: LNR



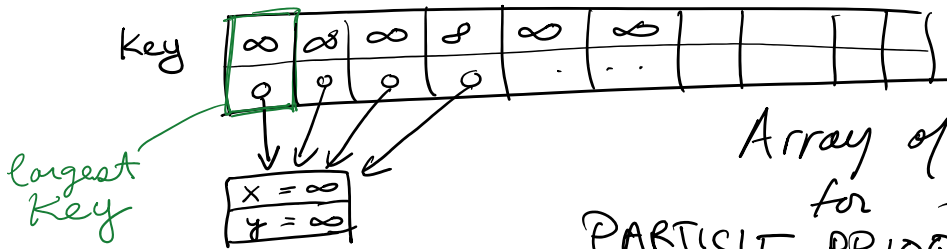
rand N rand
L N R
R N L

Closest Cell first



k=32 Initially all at ∞

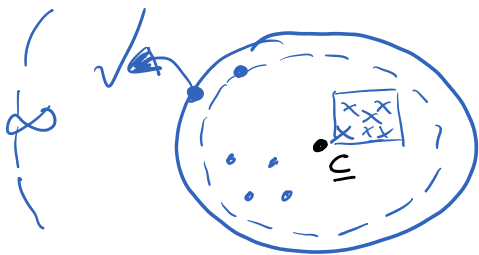
Particles which are nearest neighbors at any point



Array of 32 elements for example.

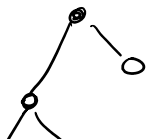
PARTICLE PRIORITY (\rightarrow)

PRIORITY QUEUES

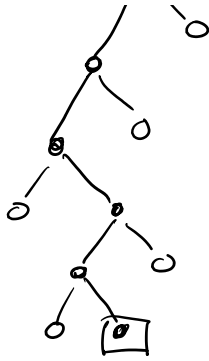


Add heuristic to speed up the search!

Traverse the tree opening the closest cells first.



This is even better than



This is even better than the following:

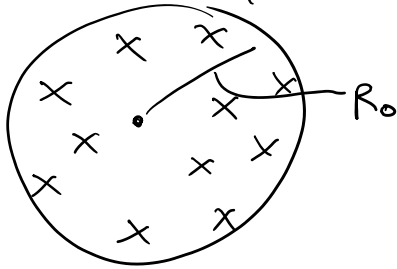
```

if (A.dist2 < B.dist2) {
    BALLSEARCH(A, ...)
    BALLSEARCH(B, ...)
} else
    BALLSEARCH(B, ...)
    BALLSEARCH(A, ...)
}

```

Finding Max/Min on the PRIORITY $O(1)$
 Insert into PRIORITY $O(N)$ Heap $\rightarrow O(\log_2 N)$

SPH



$$\rho(r) = \frac{\sum m_k}{\frac{4}{3}\pi R_0^3}$$

$$\rho(r) = \frac{\sum m_k}{\pi R_0^2} \quad \begin{matrix} \leftarrow 3\text{-D} & 2\text{D} \end{matrix}$$

