Accelerating simulations by clustering bodies using the Barnes-Hut algorithm

In a space with n-bodies, there are $n-1$ forces acting on each body. When simulating the forces acting on all the bodies, $n \cdot (n-1)$ forces need to be calculated for estimating the new position of the individual bodies. With a big enough amount of bodies, this gets problematic. Let's take a real galaxy with $2 \cdot 10^8$ Stars. The total amount of forces that need to be calculated are $4 \cdot 10^{16}$. The amount of forces that need to be calculated can be reduced by utilizing the Barnes-Hut algorithm clustering the bodies resulting in much less calculations.

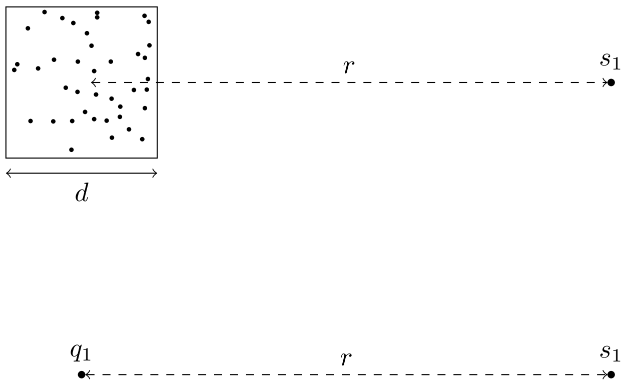
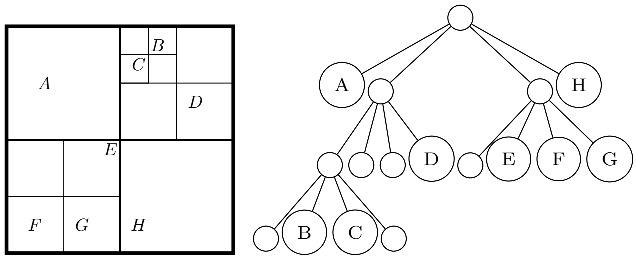


Figure 1: A cluster of stars that is far enough away from a single star can be abstracted as a single point in space.

$$\theta = \frac{d}{r} \tag{1}$$

The above equation describes how to cluster the stars. If a body is far away ($>> r$) from a small cluster ($<< d$), θ get's very small and the cluster in which the body is located can be abstracted to a single point. By defining a θ as a threshold, we can define what clusters we take into effect when calculating the forces acting on a single star. In order to do so, the space in which the objects are defined needs to be subdivided into cells. Such a subdivision can be seen in Figure **??**.

When calculating the forces on let's say the object F, not all other objects need to be taken into effect, only the ones that apply to the Barnes-Hut Principle. For the object F, this means that the Objects C and D are not calculated independently, but as one object (The midpoint of the center of gravity is defined as a new abstract object).



(a) some caption 1 (b) some caption 2

Figure 2: The cells defined in Figure **??** displayed in the form of a quad tree. (`http://arborjs.org/docs/barnes-hut`)

`@hanemile` on most platforms.

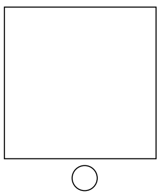


Figure 3: Caption

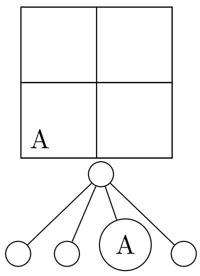


Figure 4: Caption

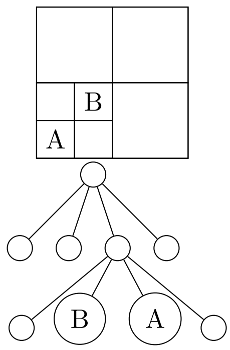


Figure 5: Caption

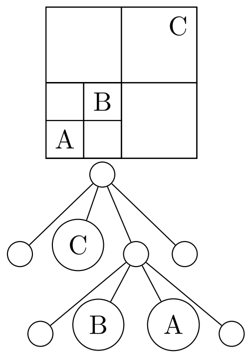


Figure 6: Caption