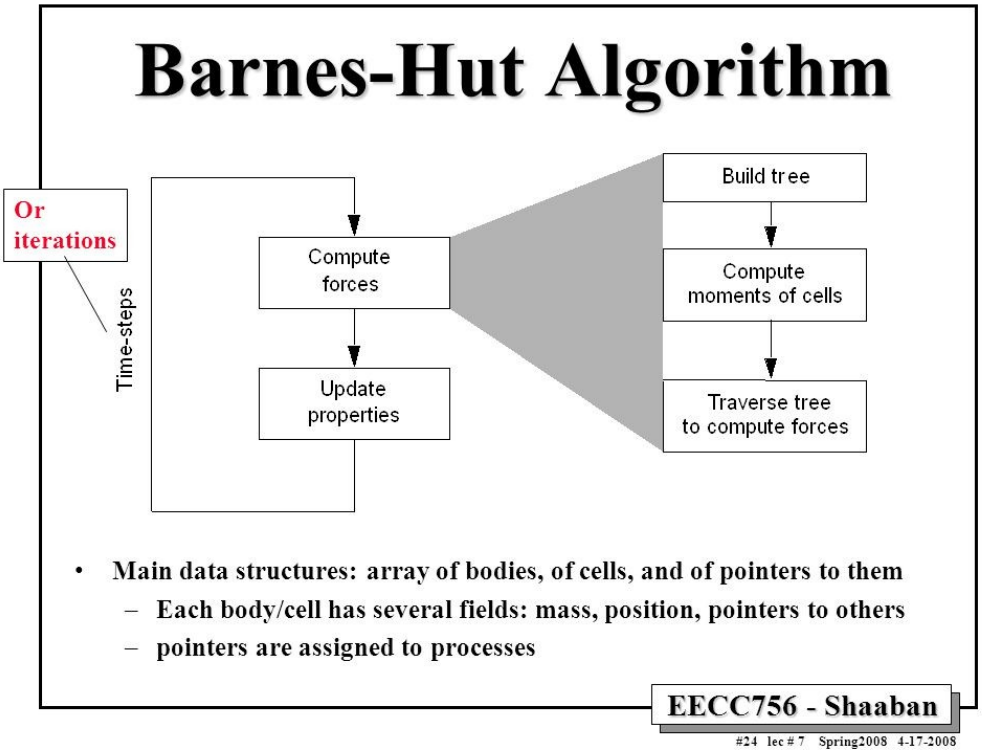
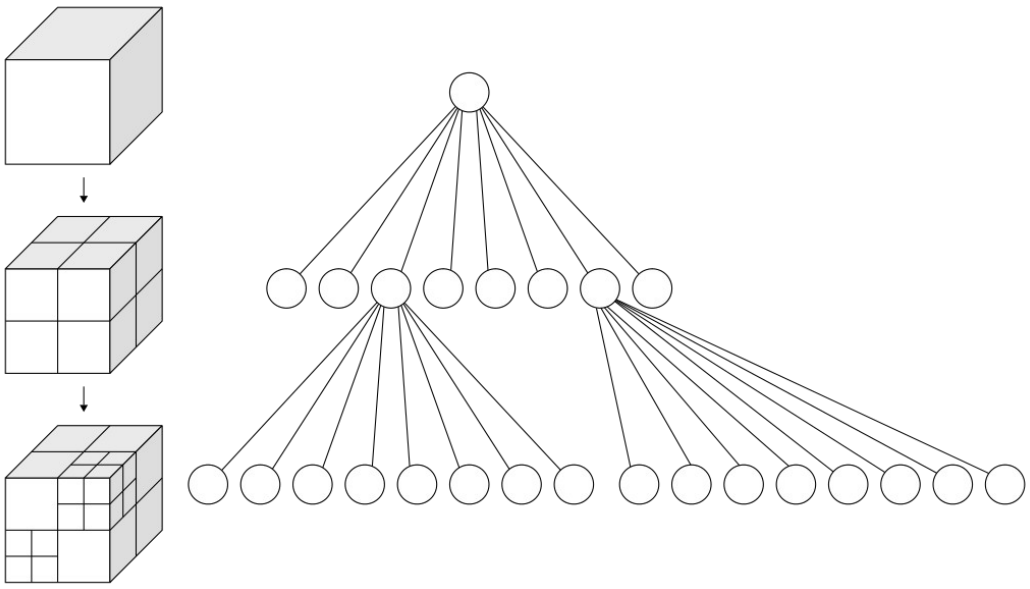
Алгоритм Барнса-Хата является приближенным алгоритмом для оптимизации некоторых задач, решаемых методом динамики частиц. Сложность алгоритма $O(n\log(n))$.



Объем делится на кубические области через **октодерево** — тип дерева (структуры данных), в которой у каждого внутреннего узла ровно восемь потомков.

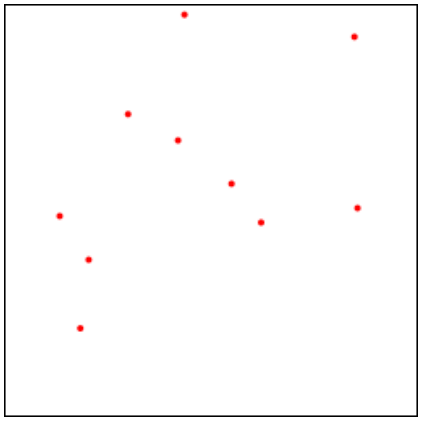
В данном алгоритме самый верхний узел представляет все пространство, а его восемь дочерних элементов представляют восемь октантов пространства. Пространство рекурсивно подразделяется на октанты, пока каждое подразделение не содержит либо 0 тел, либо 1 тело. В октодереве есть два типа узлов: *внутренние* и *внешние узлы*. Внешний узел не имеет дочерних элементов и либо пуст, либо представляет одно тело. Каждый внутренний узел содержит группу тел и хранит центр масс и общую массу всех его дочерних тел.

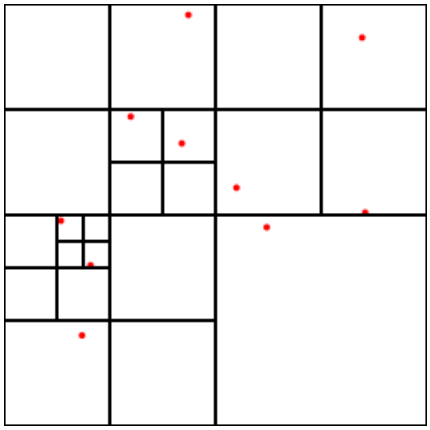


Попарно обрабатываться только частицы из соседних ячеек (во внешних узлах), а частицы в удаленных ячейках могут рассматриваться как одна большая частица с центром в центр масс

клетки. Это значительно уменьшает количество парных взаимодействий частиц, которые должны быть обработаны.

Пример разделения, но для двумерного случая (визуализация):





Реализация в коде. Кажую область можно задать, зная координаты её двух углов: далекого верхнего правого и ближнего нижнего левого. Каждый внутренний узел содержит 8 деревьев (октаны пространства). Пример кода для добавление элемента (Источник кода):

```
void Octree::add(Element e)
{
    //проверка, если это корневой или конечный элемент
    if (level == 0 || objects.empty() && is_leaf()) {
        objects.emplace_back(std::move(e));
        return;
    }
    //с каких сторон относительно центра находится элемент
    const bool left = e.pos.x < middle.x;
    const bool down = e.pos.y < middle.y;
    const bool far = e.pos.z < middle.z;
    //создание новых узлов (их 8)
    //children – массив Octree
    auto& child = children[4*left + 2*down + far];
    if (!child) {
        //определение углов нового узла (узел – область пространства)
        glm::vec3 fbl = far_bottom_left;
        glm::vec3 ntr = near_top_right;
        (left ? ntr : fbl).x = middle.x;
        (down ? ntr : fbl).y = middle.y;
        (far ? ntr : fbl).z = middle.z;
        child = std::make_unique<Octree>(fbl, ntr, level-1);
        auto to_move = std::move(objects);
        objects.clear();
        for (auto& o: to_move)
            add(std::move(o));
    }
    child->add(std::move(e));
}
```