

k-d дърво

Synopsis: Да се имплементира k-d дърво, което да поддържа построяване по списък елементи, търсене на най-близък съсед и търсене в интервал.

k-d дървото е структура от данни, която съдържа множество от точки от k-мерното пространство, откъдето идва и нейното име. Нека например пространството е \mathbb{R}^3 и представяме неговите елементи като наредени тройки (x, y, z) , които ще наричаме за простота „точки“. k-d дървото, построено по дадено множество от точки, представлява двоично дърво, където:

- Всяко листо на дървото съдържа една или повече точки от множеството.
- Всеки възел в дървото, който не е листо, символизира разбиване на множеството от точки на две подмножества. Разбиването на точките се извършва спрямо спрямо дадена ос A (една измежду осите X, Y или Z) и конкретна стойност *val* по тази ос. Тогава лявото поддърво на този възел представлява k-d дърво, съдържащо само точките от множеството с координата по оста A по-малка или равна на *val*, докато дясното поддърво е k-d дърво, съдържащо само тези точки, чиито координати по оста A са по-големи от *val*. Самият възел не съдържа точки от множеството, но може да съдържа координатите на цялото съответстващо му подпространство (напр. $x \in [25, 40]$ и $y \in [-8; 10]$), още известни като bounding box за множеството.

При построяването на k-d дървета могат да се използват множество стратегии за избиране на ос и стойност за разделяне за всеки възел.

Една възможност е възлите в последователните нива на дървото да разцепват пространствата по алтерниращи оси: коренът разцепва пространството по оста X, неговите наследници по Y, техните наследници по Z, техните по X и така нататък. За стойност за разделяне в даден възел може да се избере медианата на стойностите на точките по съответната координата на разцепване. Възможно е и оста, и стойността да се избират на произволен или псевдо-произволен принцип – например избират се 5 произволни точки от списъка и се връща медианата от техните координати по оста на разделяне.

За да не става твърде разклонено дървото, може или неговата височина да се ограничи, или да се добави възможност листата да съдържат напр. по най-много 20 точки (или комбинация от двете стратегии).

След построение k-d дърветата се използват най-често за две операции:

- търсене на най-близък съсед: по дадена точка (x_0, y_0, z_0) и метрика за разстояние между две точки в пространството да се намери тази точка, съдържаща в дървото, която е най-близо спрямо до търсената.
- търсене в интервал: по зададен интервал от стойности за всяка ос да се намерят всички съдържащи точки, чиито координати принадлежат на съответните интервали – например всички точки, за които $x \in [0, 100]$ и $y \in [-50; 50]$ и $z \in [20, 60]$.

За проекта:

Имплементирайте k-d дърво, което да поддържа тези три операции: построяване по фиксиран списък от точки, търсене на най-близък съсед и търсене в интервал. Можете за пространство да вземете \mathbb{R}^2 или \mathbb{R}^3 , но не сте задължени – ще получите бонус за по-generic решение, абстрахиращо типа точки и операциите над тях. Ще бъдат отнемани точки, ако операциите за търсене са ненужно бавни. Също така е желателно да изберете специфична стратегия за по-добро построение на дървото.

Бонуси по този проект могат да бъдат дадени освен за generic решение, за използване на рандомизирана стратегия за построяване на дървото и/или сериализиране и десериализиране във файл (формат и съдържание по ваш избор).