

1.KD-TREE with KNN search

1-1 Library:

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
```

1-2 Language: C

1-3 Environment: compiler:LLVM9.0.0 (but can also work with gcc5.4.0)

1-4 'K' I use and reason:

由於這次作業KNN中的K是被指定好的，所以我使用的K就是1,5,10,100。
理由就是因為作業指定的。

1-5 Usage:

程式的邏輯概述如下：

Build KD-tree—

判斷節點基準：從維度0開始，一直使用到維度9，然後再從維度0開始，可不斷重複。

判斷節點方法：將所有的train data按照當前判斷維度做排序(Quicksort)。排序完成後找中位數，小於等於中位數的值放入左子樹，大於的值放入右子樹。中位數用來建立節點，不往下分。

終止條件：當每個leaf node都只剩一個data時。

KNNList—

我用MaxHeap來儲存最接近的k個neighbors，若heap還沒達到k個，不論是什麼值都放入。若達到k個了，放入的值會和Heap中的第一個值比較，留下離test data比較近的那個。更改KNNList後會使用**FixUp**或**FixDown**來做修正，保持knnlist中距離test data最遠的值永遠在第一個。

Trace KD-tree—

將test data丟入，並沿著樹根往下走，直到走到leafnode為止。走到leaf後沿著parent往上找，距離較近就丟入knnlist。若現在停留的node的維度與父節點的超平面相切，及表示在另一半的子樹中有可能有更近的點，要進入另一半搜尋。搜尋方式重複**Trace KD-tree**的方法，故這邊採用recursive的寫法處理。直到走到root的父母為止。