

原创 学习图论（四）——最短路问题

2018-04-14 20:26:53 刻苦驴啊 阅读数 172 更多

版权声明：本文为博主原创文章，遵循 CC 4.0 BY-SA 版权协议，转载请附上原文出处链接和本声明。

本文链接：https://blog.csdn.net/D5_J9/article/details/79943836

学习时参考的博客：<https://blog.csdn.net/qibofang/article/details/51594673>

一、DFS或BFS搜索（单源最短路径）

思想：遍历所有从起点到终点的路径，选取一条权值最短的路径。

下面代码是参考博客中的代码，加上本人一些注释

```
1 void DFS(int u,int dist) //u 为当前节点： dist 为当前点到起点的距离
2 {
3     //min 表示目前起点到终点的最短距离，初始化为 无穷（用一个很大的数去表示）
4     if(dist>min) //当前距离已经大过目前起点终点的最短距离，没必要往下
5         return ;
6     if(u==dest) //到达目的地
7         if(min>dist)
8         {
9             min=dist;
10            return ;
11        }
12    for(int i=1;i<=n;++i) //遍历每一条边，标记点
13    {
14        //对于同一个点，edge[i][i] = 0，下面条件中让 edge[u][i]大于0，是为了防止陷入无限循环
15        //即老是搜索同一个点。
16        if(edge[u][i]!=INF&&edge[u][i]&&!vis[i])
17        {
18            vis[i]=1;
19            DFS(i,dist+edge[u][i]); //更新当前距离
20            vis[i]=0;
21        }
22    }
23    return ;
24 }
```

二、弗洛伊德算法（多源最短路径）

思想：通过比较得出图中的任意两个点，依次通过0个，1个，2个...中介点所形成的路径中最短的那一条。

一开始如果两点（U和V）之间的路径（路径①）上没有其他点，那么这条路径必定是这两点过上的最短路径。现在如果两点之间有另外一条路径（路径②）上存在第三个点（T），若这条路径比原来的路径①短，则将该路径②更新为U和V之前的最短路径，而这条路径②，是由U-T，T-V两条直接相连的路径组成，当存在一条路径连接U、T、V且上面有第四个点时，在这条路径上U到V的距离是否更小？...

核心代码：

用了三层循环，时间复杂度较高

（这篇博文对弗洛伊德算法解释很详细：https://blog.csdn.net/qq_34374664/article/details/52261672）

```
1 void Floyd(int all) //all 共有多少节点
2 {
3     //三层循环表示的是：从 j 到 k 经过点 i 后 j和k之间的距离有没有缩小
4     for(int i=1;i<=all;++i)
5         for(int j=1;j<=all;++j)
6             for(int k=1;k<=all;++k)
7                 //最好加入 egde[j][i] 和 egde[i][k] 都小于inf的条件，防止数据爆掉
8                 if(egde[j][k]>egde[j][i]+egde[i][k]) //判断借助中转点后，路程是否缩小
9                     egde[j][k]=egde[j][i]+egde[i][k]; //每个 egde[i][j]都表示当前时刻下 i 到 j 的最短距离
10 }
```

使用Floyd算法后，二维数组edge表示的边都是两点之间的最短路，所以可以输出任意两点之间最短的路径。

三、迪杰斯特拉算法（单源最短路径）

思想：通过 $n-1$ 层循环，每一次，寻找距离起点最近的点，做标记（不再访问），然后用该点作为其他点到起点路径上的中介，更新其他点到起点的距离。即，假设起点为U，中介点为T，图上任意没有被标记过的点为V，判断U-V的距离是否大于U-T-V距离，是的话更新U-V的距离为U-T-V的距离。（用一存放在图上其他点到起点的距离），这种对两点直接距离的更新称为**松弛操作**（记住一些专业名词对以后的阅读有好处）。然后再次从未标记的点中寻找，做同样的操作，知道遍历所有的点。最终得到的 `dist` 数组中存放的是图上任意点到起点的**最短距离**。

为什么可以实现找到最短路？

因为这个算法先是找到和起点直接相连的距离最短的点，在更新其他点到起点的距离的时候，都是利用已计算过和起点的最短距离的点作为中介点去比得到的必然都是最短距离。

```
1 void Dijkstra(int u)
2 {
3     int min;//表示某一点到起点的最短距离
4     int k;// k 记录为访问的点中距离起点最近的点
5     for(int i=1; i<=n; ++i)
6     {
7         flag[i]=0;// flag 用于标记节点是否已经访问过;
8         //dist 数组, 存储图上各点到起点的距离
9         dist[i]= edge[u][i]; // edge 存放的是两个节点之间的距离, 若两个点之间没有边, 则赋值为 INF (极大的数)
10        pre[i] = 0; //pre 数组用来存储 最短路径中节点 i 的上一个节点是哪一个
11    }
12    flag[u]=1;//标记起点, 没必要访问
13    dist[u]=0;
14    //开始Dijkstra算法, 遍历 n-1次, 即遍历除起点外其他所有的点
15    for(int i=1; i<n; ++i)
16    {
17        //寻找离起点最近的点
18        min = INF;
19        for(int j=1; j<=n; ++j)
20            if(!flag[j]&&min>dist[j])
21            {
22                min=dist[j];
23                k = j;
24            }
25        //相应的点后, 更新图上其他点到起点的路径距离
26        flag[k]=1;
27        for(int j=1; j<=n; ++j)
28            if(edge[k][j]!=INF)// k 和 j 之间存在边时
29            {
30                int temp = edge[k][j]+dis[k]; //i 到 j 的新路径
31                if(!flag[j]&&temp<dis[j]) //如果该路径距离小于原本 j到i 的距离, 刷新替换
32                {
33                    dis[j]=temp;
34                    pre[j]=k;
35                }
36            }
37    }
38 }
```

此处补充四个小概念：（N表示节点数，M表示边数）

- ①稀疏图：M远比 N^2 小的图称为稀疏图
- ②稠密图：M相对较大的图称为稠密图
- ③大顶堆：堆顶元素最大的堆
- ④小顶堆：堆顶元素最小的堆

紫书上对该算法进行了优化并且封装，其中涉及**优先队列**的知识，基本思想一致，用优先队列只是进行了优化，使得即使是稠密图，使用该优化算法的接矩阵的算法快。因为执行 `push` 存在前提。

（用优先队列，是因为算法中 $d[i]$ 越小，越应该先出队（每次先取和起点距离最近的点））

下面给出紫书把此算法封装成结构体的代码

链接所指大佬提供了迪杰斯特拉算法堆优化的另一种思路：不需要定义结构体重载 `<`，而是直接对距离取相反数

```
//把整个算法以及所用到的数据结构封装到一个结构体中
struct Dijkstra
{
    struct Edge
    {
        int from,to,dist;
        Edge(int u,int v,int d):from(u),to(v),dist(d) {}
    }; //存放边的结构体
```

```

int n,m;
vector<Edge> edges;
vector<int> G[MAXN];
bool done[MAXN]; // 标记是否访问过
int d[MAXN]; // 起点到各个点的距离
int p[MAXN]; // 记录最短路径

void inti(int n)
{
    this->n=n; //赋值节点数
    //清空操作,是为了可以重复使用此结构体
    for(int i=1; i<=n; ++i)
        G[i].clear();
    edges.clear();
}

void AddEdge(int f,int t,int d)
{
    edges.push_back(Edge(f,t,d));
    m = edges.size(); //每次添加边后都把边数赋值给 m 最后得到总边数
    G[f].push_back(m-1); //m-1是保存了 边在edges中的编号(下标),可以通过此编号访问边
}

//定义一个结构体作为优先队列中的元素类型
struct HeapNode{
    int d,u; //d 是距离, u 是节点
    //表示优先级为从小到大
    bool operator<(const HeapNode& a)const
    {
        return d>a.d; //小堆顶
        //个人理解,队列的每次push操作都要根据优先级判断再放置队顶元素,
        //即是否 队顶<新入队元素的操作(STL默认操作是大根堆)
        //重载<运算符后使得比较结果颠倒,优先级变成从小到大
    }
};

//主算法
void dijkstra(int u)
{
    priority_queue<HeapNode> q; //创建一个优先队列
    for(int i=1;i<=n;++i)
        d[i] = INF;
    d[u] = 0; //起点到起点的距离当然为0
    memset(done,0,sizeof(done));
    HeapNode v(0,u); //起点
    q.push(v);
    while(!q.empty())
    {
        v=q.top();
        q.pop();
        if(done[v.u]) //已经访问过的点不再访问
            continue;
        done[v.u] = true;

        //可以省去done数组,改为 if(v.d!=d[v.u]) 防止节点重复扩展

        //思路同,只是实现方式改变
        for(int i=0;i<G[v.u].size();++i) //这种情况下每条边只被访问了一次
        {
            Edge& e = edges[G[v.u][i]];
            if(d[e.to]>d[v.u]+e.dist)
            {
                d[e.to]=d[v.u]+e.dist;
                p[e.to]=v.u;
                q.push((HeapNode){d[e.to],e.to});
            }
        }
    }
}

};

```

注意: 迪杰斯特拉算法不可以求含有负权边的最短路!
具体原因等我搞明白了再写一篇博客。

四、Bellman-Ford (贝尔曼 - 福特) 算法 (单源最短路径, 可以求解含负权边的图)

思想: (节点数 N , 边数 M) 进行至多 $N-1$ 次循环, 每一次都更新图上任意点到起点的距离 (松弛操作)。其实个人觉得类似Dijkstra算法, 对任意点到进行更新, 只是每一次都遍历所有边去更新。至多 $N-1$ 次, 是考虑了最坏的情况, 即每一次遍历所有边, 只有和起点相连的边得到更新。

核心代码:

```
1 // u[i]和v[i] 分别存放第 i 条边的前后顶点, w[i]存放权值
2 bool BellmanFord()
3 {
4     for(int i=1;i<=n;++i)
5         d[i]=(i==1?0:INF);
6     //至多遍历 n-1 次
7     //考虑最坏的情况, 即每一次遍历所有边, 只有和起点相连的边得到更新
8     for(int i=1;i<=n;++i)
9         for(int j=1;j<=m;++j)
10            {
11                int x=u[i],y=v[i];
12                if(d[x]<INF)
13                    d[y]=min(d[y],d[x]+w[i]);
14            }
15     //判断是否存在 含有负权的环
16     for(int i=1;i<=m;++i)
17         if(d[v[i]]>d[u[i]]+w[i])
18             return false;
19     return true;
20 }
```

参考此博客 (<https://www.cnblogs.com/xiu68/p/7993514.html>)

此算法的缺点是效率不高。因为要进行 $N-1$ 次循环, 但可能在第一次就已经得到答案了, 亦即是做了很多无用的工作。

下面是对该算法的改进, 一般称为SPFA算法。

思想: 在Bellman-Ford算法的基础上, 使用队列保存待松弛的点, 对每个出队的点, 遍历跟它相邻的点, 如果这两点间的边能够松弛且相邻点不在队列邻点存入队列。(因为松弛操作可能对该相邻点的相邻点产生影响)

核心代码:

```
1 void SPFA()
2 {
3
4     vector<int> d(n+1,INF); //距离
5     vector<int> cnt(n+1,0); //记录每个点遍历的次数
6     vector<int> inq(n+1,0); //记录节点是否在队列内
7     d[s]=0;
8     cnt[s]=inq[s]=1;
9     deque<int> q(1,s);
10    while(!q.empty())
11    {
12        int u=q.front(),i=0,to,dist;
13        inq[u] = 0; //出队
14        q.pop_front();
15        for(; i<g[u].size(); ++i)
16        {
17            to=g[u][i].to;//g[u][i].to表示u-i边箭头端 (末端) 的点
18            dist=g[u][i].dist;
19            if(d[to]>d[u]+dist)
20            {
21                d[to]=d[u]+dist;
22                if(!inq[to]) //如果点不在队列内再入队
23                {
24                    //若有负权环, 则会无限循环下去, 所以肯定有一个点遍历的次数大于 n
25                    if(n==++cnt[to])return 0; //判断是否为负权环
26                    inq[to]=1;
27                    //SLF优化: 减少重复扩展的次数。
28                    if(!q.empty()&&d[to]<q.front()) //确保队列非空才能访问q.front()
29                        q.push_front(to);
30                    else
31                        q.push_back(to);
32                }
33            }
34        }
35    }
36 }
```

其中用了SLF优化（关于此优化，好像还没有确切的证明，但实际运用上确实优化了不少，如果有读者知晓麻烦评论告知一下，多谢）
SLF优化：判断将要入队的节点到源点的距离是否比队首元素小，是的话插入队首，否则插入队尾。所以上面代码用了双向队列。这样做可能会减少原
接点或者一些本来没有SLF优化时需要入队的点的入队次数，即减少了扩展次数。但是存在反例使得使用了SLF优化反而使程序变慢。所以一般可以用，
都能用。（我对此的理解还很浅薄）

学好图论的路，还很长...

文章最后发布于: 2018

有 0 个人打赏

最短路四种算法浅理解(大佬请无视)

阅读数 134

第一种~~~~~dijkstra~~~~(加边法)这个题目就是一个dijkstra... 博文 来自: ecjtu_17_...

想对作者说点什么

图论基础——最短路问题

阅读数 583

一、单源最短路问题1。（Bellman-Ford）1.当图为DAG时，把图拓扑排序一下，然... 博文 来自: 叶子心情...

图论初步——最短路(2)

阅读数 47

Dijkstra算法优化关于Dijkstra详见图论初步——最短路(1)前面的文章中，我们提到了... 博文 来自: wljoi的博客

图论初步——最短路(1)

阅读数 54

最短边分类单源最短路：从一个点到其他所有点的最短路算法：Dijkstra,spfaDijkstra,... 博文 来自: wljoi的博客

高一数学向量知识点

高等数学补课

1963阅读

图论——最短路

阅读数 84

图论中的常见问题，有三种常用算法，以及许多拓展内容。松弛操作对于每条有向边(... 博文 来自: 又又大柚...

图论2——最短路

阅读数 18

【概述】最短路是图论中十分常见的问题，可分为单源最短路与全源最短路。对... 博文 来自: Alex_Mc...

图论——最短路,最小生成树。

阅读数 2

最短路： Dijkstra：每次拓展当前未拓展的最近点X，因为X不可能被再次更新， ... 博文 来自: weixin_3...

图论——最短路——Floyd 算法

阅读数 693

【概述】Floyd算法又称为插点法，是一种用于寻找给定的加权图中多源点之间最短路... 博文 来自: Alex_Mc...

图论——最短路算法学习笔记

阅读数 22

最短路算法1.FLOYD算法多源最短路预处理：二维数组储存两点之间的边距离，初始... 博文 来自: Sensente...

学习图论(四)——最短路问题 - CSDN博客

基础最短路四 POJ3268 - Hide_in_Code - CSDN博客

年度榜单：Python三连冠，碾压Java！你怎么看？

IEEE Spectrum近日发布了2019年度编程语言排行榜，令人些许意外的是，Python连续三年问鼎巅峰，你怎么看？

图论——最短路——Johnson 算法

阅读数 25

【概述】对于单源最短路来说，有时间复杂度为O(E+VlogV)要求权值非负的Dijkstra... 博文 来自: Alex_Mc...

极其简单的最短路问题【BFS】【图论】 - ! - CSDN博客

最短路学习记录 - qq_40723554的博客 - CSDN博客



图论 —— 最短路 —— Dijkstra 算法

阅读数 579

【概述】Dijkstra算法是单源最短路径算法，即计算起点只有一个的情况到其他点的最... 博文 来自: Alex_Mc...



信天翁_

54篇文章

关注 排名:千里之外



叶子心情你不懂

131篇文章

关注 排名:千里之外



NDMZ_WLJ55

43篇文章

关注 排名:千里之外



又又大柚纸

42篇文章

关注 排名:千里之外

【BFS】【图论】极其简单的最短路问题 - giaogiao - CSDN博客

图论总结(6)最短路问题 - qq_30802053的博客 - CSDN博客

图论算法——最短路系列

阅读数 4

最短路经典算法整理1.弗洛伊德：三重循环例题：codevs1020孪生蜘蛛时间限制:1s空... 博文 来自: weixin_3...

图论——最短路——算法（3.0）

阅读数 234

一、只有5行代码的floyd算法：1、什么是floyd算法弗洛伊德算法是解决多元最短路... 博文 来自: T.J的博客

知识点四 图论:Floyd(任意两点最短路) - Authur_gyc - CSDN博客

最短路问题【图论】【最短路】(四种方法) - 菜鸡的bl... - CSDN博客

图论基础算法——最短路之Dijkstra算法

阅读数 251

1.单源最短路：介绍Dijkstra算法之前先介绍单源最短路的概念吧!而Dijkstra算法常常用... 博文 来自: ly's Blog

反转!“只问了1个框架，就给了35K的Python岗”

学Python的程序员建议收藏!



[数模笔记]图论-最短路问题 - 济川的博客 - CSDN博客

图论最短路例题总结&&分享做法 - 一只酷酷光儿的博客 - CSDN博客

图论 —— 最短路 —— Bellman-Ford 算法与 SPFA

阅读数 405

【概述】Bellman-Ford算法适用于计算单源最短路径，即：只能计算起点只有一个的... 博文 来自: Alex_Mc...

图论——最短路——Dijkstra算法

阅读数 14

对图论有一定了解的人，一定知道最短路。最短路算法一共有4中，严格来说是3种， ... 博文 来自: weixin_3...

图论——最短路之Floyd&Dijkstra

阅读数 83

概念： 求图上一点到另一点的最短距离。算法： 一：Floyd（弗洛伊德） Floy... 博文 来自: chenkain...

程序员实用工具网站

阅读数 17万+

目录1、搜索引擎2、PPT3、图片操作4、文件共享5、应届生招聘6、程序员面试题库... 博文 来自: 不脱发的...

程序员真是太太太太有趣了!!!

阅读数 5万+

网络上虽然已经有了很多关于程序员的话题，但大部分人对这个群体还是很陌生。我... 博文 来自: 程序猿DD

那些拿到 60K Offer 的 AI 程序员，后来都怎么样了？

刚刚拿到阿里offer，工作地点杭州。值得去吗？

1行Python代码制作动态二维码

阅读数 1万+

目录1、普通二维码2、艺术二维码3、动态二维码在GitHub上发现了一个比较有意思... 博文 来自: 不脱发的...

全球最厉害的 14 位程序员!

阅读数 8273

来源|ITWorld整理自网络全球最厉害的14位程序员是谁?今天就让我们一起来了解一... 博文 来自: GitHubD...



从入门到精通，Java学习路线导航

阅读数 6万+

引言最近也有很多人来向我“请教”，他们大都是一些刚入门的新手，还不了解这个行... 博文 来自： wangweij...

我花了一夜用数据结构给女朋友写个H5走迷宫游戏

阅读数 15万+

起因又到深夜了，我按照以往在csdn和公众号写着数据结构！这占用了我大量的时间... 博文 来自： bigsai

别再翻了，面试二叉树看这 11 个就够了~

阅读数 6万+

写在前边数据结构与算法：不知道你有没有这种困惑，虽然刷了很多算法题，当我去... 博文 来自： 一个不甘...

从零开始深度学习102讲，送配套图书！

从原理到实战，美女科学家带你入门深度学习

接班马云的为何是张勇？

阅读数 3万+

上海人、职业经理人、CFO背景，集齐马云三大不喜欢的张勇怎么就成了阿里接班人... 博文 来自： CSDN资讯

什么是大公司病（太形象了）

阅读数 7556

点击蓝色“五分钟学算法”关注我哟加个“星标”，天天中午12:15，一起学算法作者... 博文 来自： 程序员吴...

让程序员崩溃的瞬间（非程序员勿入）

阅读数 21万+

今天给大家带来点快乐，程序员才能看懂。来源：https://zhuanlan.zhihu.com/p/47... 博文 来自： stronger...

离职了

阅读数 4005

这是我毕业后的第一份工作...面试时，HR小姐姐告诉我...然鹅...我入职之后才发现：对... 博文 来自： 嵌入式Lin...

iPhone 11 引领芯片新革命？

阅读数 986

iPhone11中的U1芯片开启了超宽带革命。作者|JasonSnell译者|弯月，责编|郭芮出品... 博文 来自： CSDN资讯

[推动全社会公益氛围形成，使公益与空气和阳光一样触手可及。](#)

公益缺你不可，众多公益项目等你PICK——百度公益 让公益像「空气和阳光」一样触手可及！
gongyi.baidu.com

如何在Windows中开启"上帝模式"

阅读数 8894

原文链接:https://mp.weixin.qq.com/s?__biz=MzlwMjE1MjMyMw==&mid=... 博文 来自： 时间静止

分享靠写代码赚钱的一些门路

阅读数 5万+

作者mezod，译者josephchang10如今，通过自己的代码去赚钱变得越来越简单，不... 博文 来自： Python之...

失业42天，我废了

阅读数 1538

作者：子或师兄https://www.jianshu.com/p/62590c1339f12019.6.5这天下午，公... 博文 来自： Python之...

技术人员要拿百万年薪，必须要经历这9个段位

阅读数 1万+

很多人都问，技术人员如何成长，每个阶段又是怎样的，如何才能走出当前的迷茫，... 博文 来自： Python之...

8000字干货：那些很厉害的人是怎么构建知识体系的

阅读数 3万+

本文约8000字，正常阅读需要15~20分钟。读完本文可以获得如下收益：分辨知识和... 博文 来自： 程序视界...

一本可陪伴一辈子的笔记本

可擦可写，可循环利用，支持OCR识别，让你的笔记本变得智能

分布式、多线程、高并发都不懂，拿什么去跳槽

阅读数 1万+

当提起这三个词的时候，是不是很多人都认为分布式=高并发=多线程？当面试官问到... 博文 来自： Java知音

nginx学习，看这一篇就够了：下载、安装、使用：正向代理、反向代理、...

阅读数 4272

文章目录前言一、nginx简介1.什么是nginx和可以做什么事情2.Nginx作为web服务器... 博文 来自： 冯安晨

动画：用动画给面试官解释 TCP 三次握手过程

阅读数 3万+

作者|小鹿来源|公众号：小鹿动画学编程写在前边TCP三次握手过程对于面试是必考的... 博文 来自： 一个不甘...

为什么程序员在学习编程的时候什么都记不住？

阅读数 3万+

在程序员的职业生涯中，记住所有你接触过的代码是一件不可能的事情！那么我们该... 博文 来自： CSDN资讯



500行代码，教你用python写个微信飞机大战

阅读数 5万+

这几天在重温微信小游戏的飞机大战，玩着玩着就在思考人生了，这飞机大战怎么就...

博文 来自: [Python专栏](#)

一张图看懂嵌入式系统组成

阅读数 751

嵌入式系统是一种应用范围非常广泛的系统。可以说除了一般用途的计算机外的所有...

博文 来自: [不脱发的...](#)

学会了这些技术，你离BAT大厂不远了

阅读数 14万+

每一个程序员都有一个梦想，梦想着能够进入阿里、腾讯、字节跳动、百度等一线互...

博文

记一道字节跳动的算法面试题

阅读数 1万+

点击蓝色“五分钟学算法”关注我哟加个“星标”，天天中午 12:15，一起学算法作...

博文

史上最详细的IDEA优雅整合Maven+SSM框架（详细思路+附带源码）

阅读数 2万+

网上很多整合SSM博客文章并不能让初探ssm的同学思路完全的清晰，可以试着关掉...

博文

Python爬取淘宝商品信息

阅读数 7715

各位同学们，好久没写原创技术文章了，最近有些忙，所以进度很慢。警告：本教程...

博文



人脸识别主要算法原理

人脸识别算法

4003阅读

盘点那些被AI换脸、一键“脱”衣所滥用的AI模型

阅读数 9954

上周作者发布了一篇有关AI换脸的教程，不过令笔者始料未及的是一石激起千层浪，...

博文

五分钟小知识：为什么说 ++i 的效率比 i++ 高？

阅读数 2506

点击蓝色“五分钟学算法”关注我哟加个“星标”，天天中午 12:15，一起学算法作...

博文

工厂模式，从第三方登录说起

阅读数 4814

现在的很多平台在登陆的时候，下面都会有一排选项，可以选择微信、QQ、微博账号...

博文

什么是“中台”？

阅读数 774

“中台”这个概念，越来越多的在各种技术大会上提及，各大技术公司，纷纷推出自...

博文

为什么面向对象糟透了？

阅读数 3万+

又是周末，编程语言“三巨头”Java, Lisp 和C语言在Hello World咖啡馆聚会。服务...

博文



cdn及cdn加速原理

海外cdn加速专线

2.9万阅读

顶级产品经理是如何利用王者荣耀，3步毁掉你的自律。

阅读数 9250

【老王提示】：本文共 2384 字数，预计阅读时间为 8 Minute。前言 当今时代，王...

博文

JAVA实现商品信息管理系统

阅读数 2918

任务与实现 超市商品管理系统 题目要求 超市中商品分为四类，分别是食品、化妆品...

博文

2019诺贝尔经济学奖得主：贫穷的本质是什么？

阅读数 1万+

2019年诺贝尔经济学奖，颁给了来自麻省理工学院的 阿巴希·巴纳吉（Abhijit Vinaya...

博文

linux：最常见的linux命令（centOS 7.6）

阅读数 1万+

最常见，最频繁使用的20个基础命令如下： 皮一下，这都是干货偶，大佬轻喷 一、li...

博文

只因写了一段爬虫，公司200多人被抓！

阅读数 10万+

“一个程序员写了个爬虫程序，整个公司200多人被端了。” “不可能吧！” 刚从朋...

博文

别在学习框架了，那些让你起飞的计算机基础知识。

阅读数 5万+

我之前里的文章，写的大部分都是与计算机基础知识相关的，这些基础知识，就像我...

博文



java学习路线导航【教学视频+博客+书籍整理】
在博主认为，学习java的最佳学习方法莫过于视频+博客+书籍+总结，前三者博主将...

阅读数 7613
[博文](#)

五款高效率黑科技神器工具，炸裂好用，省时间
loonggg读完需要4分钟速读仅需2分钟感觉我好久好久没有给大家分享高质量的软件...

阅读数 2万+
[博文](#)

程序员必须掌握的核心算法有哪些？
由于我之前一直接调数据结构以及算法学习的重要性，所以就有一些读者经常问我，...

阅读数 6万+
[博文](#)

SQL基本语法入门 看这里就够了
SQL执行顺序 第一步：执行FROM 第二步：WHERE条件过滤 第三步：GROUP BY 分...

阅读数 4584
[博文](#)

如何优化MySQL千万级大表，我写了6000字的解读
这是学习笔记的第2138篇文章 千万级大表如何优化，这是一个很有技术含量的问题，...

阅读数 3万+
[博文](#)

面试最后一问：你有什么问题想问我吗？
尽管，我们之前分享了这么多关于面试的主题： 高薪必备的一些Spring Boot高级面...

阅读数 3万+
[博文](#)

python 程序员进阶之路：从新手到高手的100个模块
在知乎和CSDN的圈子里，经常看到、听到一些 python 初学者说，学完基础语法后，...

阅读数 4万+
[博文](#)

Python——画一棵漂亮的樱花树（不同种樱花+玫瑰+圣诞树喔）
最近翻到一篇知乎，上面有不少用Python（大多是turtle库）绘制的树图，感觉很漂...

阅读数 3万+
[博文](#)

Linux/C/C++ 不可错过的好书
来源：公众号【编程珠玑】 作者：守望先生 ID：shouwangxiansheng 前言 经常有...

阅读数 8171
[博文](#)

史上最强Tomcat8性能优化
文章目录授人以鱼不如授人以渔目的服务器资源Tomcat配置优化Linux环境安装运行T...

阅读数 2万+
[博文](#)

单点登录（SSO）
一、SSO（单点登录）介绍 SSO英文全称Single SignOn，单点登录。SSO是在多个...

阅读数 1万+
[博文](#)

漫话：什么是 https ?这应该是全网把 https 讲的最好的一篇文章了
今天这篇文章，讲通过对话的形式，让你由浅入深着知道，为什么 Https 是安全的。...

阅读数 2万+
[博文](#)

史上最全的mysql基础教程
启动与停止 启动mysql服务 sudo /usr/local/mysql/support-files/mysql.server sta...

阅读数 9898
[博文](#)

为什么你学不会递归？告别递归，谈谈我的经验
可能很多人在大一的时候，就已经接触了递归了，不过，我敢保证很多人初学者刚开...

阅读数 1万+
[博文](#)

大学四年，分享看过的优质书籍
数据结构与算法是我在大学里第一次接触到的，当时学了很多其他安卓、网页之类的...

阅读数 1万+
[博文](#)

有哪些让程序员受益终生的建议
从业五年多，辗转两个大厂，出过书，创过业，从技术小白成长为基层管理，联合几...

阅读数 1万+
[博文](#)

最近程序员频繁被抓，如何避免面向监狱编程！？
最近，有关程序员因为参与某些项目开发导致被起诉，甚至被判刑的事件发生的比较...

阅读数 5万+
[博文](#)

一文搞懂什么是TCP/IP协议
什么是TCP/IP协议? 计算机与网络设备之间如果要相互通信,双方就必须基于相同的方...

阅读数 1万+
[博文](#)

大学四年自学走来，这些私藏的实用工具/学习网站我贡献出来了
大学四年，看课本是不可能一直看课本的了，对于学习，特别是自学，善于搜索网上...

阅读数 9万+
[博文](#)

学习 Java 应该关注哪些网站？
经常有一些读者问我：“二哥，学习 Java 应该关注哪些网站？” ，我之前的态度一直...

阅读数 1万+
[博文](#)

如果有人问你 MySql 怎么存取 Emoji，把这篇文章扔给他
01、前言 Emoji 在我们生活中真的是越来越常见了，几乎每次发消息的时候不带个 E...

阅读数 3295
[博文](#)


0























哪些 Java 知识不需要再学了

张无忌在学太极拳的时候，他爹的师父张三丰告诉他一定要把之前所学的武功全忘掉。更有限时礼包，买域名腾讯云解析、建站模版、1年域名型SSL证书

腾讯云新用户域名抢购 1元/年起

阅读数 1万+

广告 X

大学四年，我把私藏的自学「学习网站/实用工具」都贡献出来了

在分享之前，先说说初学者如何学习编程，这个话题想必非常的重要，要学好编程，...

阅读数 3万+

博文

中国麻将：世界上最早的区块链项目

中国麻将：世界上最早的区块链项目 最近区块链这个玩意又被市场搞的很是火热，相...

阅读数 3万+

博文

比特币原理详解

一、什么是比特币 比特币是一种电子货币，是一种基于密码学的货币，在2008年11月...

阅读数 1万+

博文

c# mvc 上传 文件 c# 扫描软件 c# 文字打印左右反转 c# byte转换成数字 c# 音量调节组件 c# wpf 界面 c# 读取证书文件的内容 c# 单例模式 工厂模式 c# dgv 树结构 c# 继承 反序列化

没有更多推荐了，返回首页

👍
0

🔗

💬

🔖

📱

<

>

赏

脉

🔊

🔔

^