

2017 年 6 月
4 日算法讲解

图

二叉树

黑白图像、迷宫

数据结构

队列

栈

并查集（不相交集类）

堆

算法

广度优先搜索（BFS）

深度优先搜索（DFS）

回溯

插入排序

选择排序

冒泡排序

箱排序

归并排序

快速排序

堆排序

高精度算法

拓扑排序

最小生成树

最短路算法

动态规划

2017 年 6 月 4 日算法讲解

2017 年 6 月 4 日

目录

2017 年 6 月
4 日算法讲解

图

二叉树

黑白图像、迷宫

数据结构

队列

栈

并查集（不相交集类）

堆

算法

广度优先搜索（BFS）

深度优先搜索（DFS）

回溯

插入排序

选择排序

冒泡排序

箱排序

归并排序

快速排序

堆排序

高精度算法

拓扑排序

最小生成树

最短路算法

动态规划

1 图

- 二叉树
- 黑白图像、迷宫

2 数据结构

- 队列
- 栈
- 并查集（不相交集类）
- 堆

3 算法

- 广度优先搜索（BFS）
- 深度优先搜索（DFS）
- 回溯
- 插入排序
- 选择排序
- 冒泡排序
- 箱排序
- 归并排序

二叉树

2017 年 6 月
4 日算法讲解

图

二叉树

黑白图像、迷宫

数据结构

队列

栈

并查集（不相交集类）

堆

算法

广度优先搜索（BFS）

深度优先搜索（DFS）

回溯

插入排序

选择排序

冒泡排序

箱排序

归并排序

快速排序

堆排序

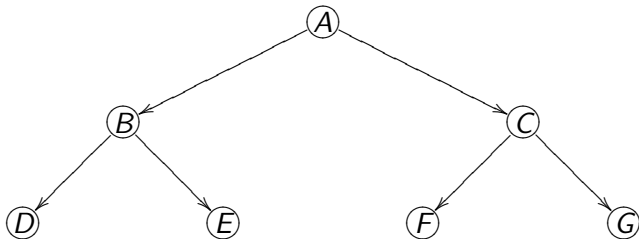
高精度算法

拓扑排序

最小生成树

最短路算法

动态规划



队列

2017 年 6 月
4 日算法讲解

图

二叉树

黑白图像、迷宫

数据结构

队列

栈

并查集（不相交集类）

堆

算法

广度优先搜索（BFS）

深度优先搜索（DFS）

回溯

插入排序

选择排序

冒泡排序

箱排序

归并排序

快速排序

堆排序

高精度算法

拓扑排序

最小生成树

最短路算法

动态规划

- 基本原理

三个基本功能：*push()*, *pop()*, *front()*

- STL 库中 `queue` 的使用

并查集（不相交集类）

2017 年 6 月
4 日算法讲解

图

二叉树

黑白图像、迷宫

数据结构

队列

栈

并查集（不相交集类）

堆

算法

广度优先搜索（BFS）

深度优先搜索（DFS）

回溯

插入排序

选择排序

冒泡排序

箱排序

归并排序

快速排序

堆排序

高精度算法

拓扑排序

最小生成树

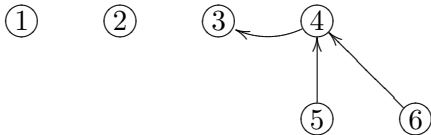
最短路算法

动态规划

union（合并树）



将 4 和 3 合并



并查集（不相交集类）

2017 年 6 月
4 日算法讲解

图

二叉树

黑白图像、迷宫

数据结构

队列

栈

并查集（不相交集类）

堆

算法

广度优先搜索（BFS）

深度优先搜索（DFS）

回溯

插入排序

选择排序

冒泡排序

箱排序

归并排序

快速排序

堆排序

高精度算法

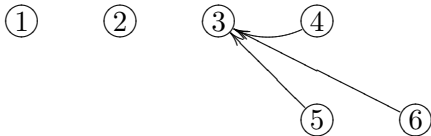
拓扑排序

最小生成树

最短路算法

动态规划

- find 查找父亲结点
递归 + 扁平化



堆

2017 年 6 月
4 日算法讲解

图

二叉树

黑白图像、迷宫

数据结构

队列

栈

并查集（不相交集类）

堆

算法

广度优先搜索（BFS）

深度优先搜索（DFS）

回溯

插入排序

选择排序

冒泡排序

桶排序

归并排序

快速排序

堆排序

高精度算法

拓扑排序

最小生成树

最短路算法

动态规划

- **定义:**

堆数据结构是一种数组对象，它可以被视为一棵完全二叉树结构。它的特点是父节点的值大于（小于）两个子节点的值（分别称为大顶堆和小顶堆）。它常用于管理算法执行过程中的信息，应用场景包括堆排序，优先队列等。

- **操作:**

`push()`, `front()`, `top()`

- **STL 库:**

`make_heap()`, `priority_queue()`

两种遍历算法

2017 年 6 月
4 日算法讲解

图

二叉树

黑白图像、迷宫

数据结构

队列

栈

并查集（不相交集类）

堆

算法

广度优先搜索（BFS）

深度优先搜索（DFS）

回溯

插入排序

选择排序

冒泡排序

箱排序

归并排序

快速排序

堆排序

高精度算法

拓扑排序

最小生成树

最短路算法

动态规划

- 广度优先搜索（BFS）
- 深度优先搜索（DFS）& 回溯

基础排序算法

2017 年 6 月
4 日算法讲解

图

二叉树

黑白图像、迷宫

数据结构

队列

栈

并查集（不相交集类）

堆

算法

广度优先搜索（BFS）

深度优先搜索（DFS）

回溯

插入排序

选择排序

冒泡排序

箱排序

归并排序

快速排序

堆排序

高精度算法

拓扑排序

最小生成树

最短路算法

动态规划

- 插入排序
- 选择排序
- 冒泡排序
- 箱排序

排序算法进阶

2017 年 6 月
4 日算法讲解

图

二叉树

黑白图像、迷宫

数据结构

队列

栈

并查集（不相交集类）

堆

算法

广度优先搜索（BFS）

深度优先搜索（DFS）

回溯

插入排序

选择排序

冒泡排序

箱排序

归并排序

快速排序

堆排序

高精度算法

拓扑排序

最小生成树

最短路算法

动态规划

- 归并排序
- 快速排序
- 堆排序

高精度算法

2017 年 6 月
4 日算法讲解

图

二叉树

黑白图像、迷宫

数据结构

队列

栈

并查集（不相交集类）

堆

算法

广度优先搜索（BFS）

深度优先搜索（DFS）

回溯

插入排序

选择排序

冒泡排序

箱排序

归并排序

快速排序

堆排序

高精度算法

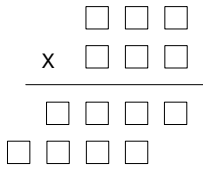
拓扑排序

最小生成树

最短路算法

动态规划

高精度加法



高精度乘法



拓扑排序

2017 年 6 月
4 日算法讲解

图

二叉树

黑白图像、迷宫

数据结构

队列

栈

并查集（不相交集类）

堆

算法

广度优先搜索（BFS）

深度优先搜索（DFS）

回溯

插入排序

选择排序

冒泡排序

箱排序

归并排序

快速排序

堆排序

高精度算法

拓扑排序

最小生成树

最短路算法

动态规划

- (1) 将所有点加入到集合中
- (2) 每次遍历全部的点，选择入度为零的点入队
- (3) 将与对头相连的的边全部删除
- (4) 重复 (2) 和 (3)

最小生成树

2017 年 6 月
4 日算法讲解

图

二叉树

黑白图像、迷宫

数据结构

队列

栈

并查集（不相交集类）

堆

算法

广度优先搜索（BFS）

深度优先搜索（DFS）

回溯

插入排序

选择排序

冒泡排序

箱排序

归并排序

快速排序

堆排序

高精度算法

拓扑排序

最小生成树

最短路算法

动态规划

- 普里姆算法 (Prim 算法)
- 克鲁斯卡尔算法 (Kruskal 算法)

2017 年 6 月
4 日算法讲解

图

二叉树

黑白图像、迷宫

数据结构

队列

栈

并查集（不相交集类）

堆

算法

广度优先搜索（BFS）

深度优先搜索（DFS）

回溯

插入排序

选择排序

冒泡排序

箱排序

归并排序

快速排序

堆排序

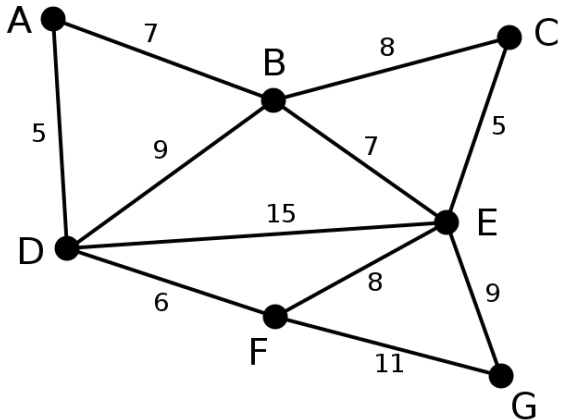
高精度算法

拓扑排序

最小生成树

最短路算法

动态规划



Prim 算法

2017 年 6 月
4 日算法讲解

图

二叉树

黑白图像、迷宫

数据结构

队列

栈

并查集（不相交集类）

堆

算法

广度优先搜索（BFS）

深度优先搜索（DFS）

回溯

插入排序

选择排序

冒泡排序

桶排序

归并排序

快速排序

堆排序

高精度算法

拓扑排序

最小生成树

最短路算法

动态规划

- ① 输入：一个加权连通图，其中顶点集合为 V ，边集合为 E ；
- ② 初始化： $V_{\text{new}} = x$ ，其中 x 为集合 V 中的任一节点（起始点）， $E_{\text{new}} = \varnothing$ ，为空；
- ③ 重复下列操作，直到 $V_{\text{new}} = V$ ：
 - ① 在集合 E 中选取权值最小的边 $\langle u, v \rangle$ ，其中 u 为集合 V_{new} 中的元素，而 v 不在 V_{new} 集合当中，并且 $v \in V$ （如果存在有多条满足前述条件即具有相同权值的边，则可任意选取其中之一）；
 - ② 将 v 加入集合 V_{new} 中，将 $\langle u, v \rangle$ 边加入集合 E_{new} 中；
- ④ 输出：使用集合 V_{new} 和 E_{new} 来描述所得到的最小生成树。

Kruskal 算法

2017 年 6 月
4 日算法讲解

图

二叉树

黑白图像、迷宫

数据结构

队列

栈

并查集（不相交集类）

堆

算法

广度优先搜索（BFS）

深度优先搜索（DFS）

回溯

插入排序

选择排序

冒泡排序

箱排序

归并排序

快速排序

堆排序

高精度算法

拓扑排序

最小生成树

最短路算法

动态规划

- ① 记 $Graph$ 中有 v 个顶点， e 个边
- ② 新建图 $Graph_{new}$ ， $Graph_{new}$ 中拥有原图中相同的 e 个顶点，但没有边
- ③ 将原图 $Graph$ 中所有 e 个边按权值从小到大排序
- ④ 循环：从权值最小的边开始遍历每条边直至图 $Graph$ 中所有的节点都在同一个连通分量中
if 这条边连接的两个节点于图 $Graph_{new}$ 中不在同一个连通分量中
添加这条边到图 $Graph_{new}$ 中

Dijkstra 算法

2017 年 6 月
4 日算法讲解

图

二叉树

黑白图像、迷宫

数据结构

队列

栈

并查集（不相交集类）

堆

算法

广度优先搜索（BFS）

深度优先搜索（DFS）

回溯

插入排序

选择排序

冒泡排序

桶排序

归并排序

快速排序

堆排序

高精度算法

拓扑排序

最小生成树

最短路算法

动态规划

Dijkstra 算法是一种用来处理边权为正且无环的有向或无向图的单源最短路算法。

伪代码：

- ① 清除所有点的标号
- ② 设 $d[0]=0$ ，其他 $d[i]=\text{INF}$
- ③ 循环 n 次 {
 - 在所有未标号节点中，选出 d 值最小的节点 x
 - 给节点 x 标记
 - 对于从 x 出发的所有边 (x, y) ，更新 $d[y] = \min(d[y], d[x] + w(x, y))$}

Dijkstra 算法的优化

2017 年 6 月
4 日算法讲解

图

二叉树

黑白图像、迷宫

数据结构

队列

栈

并查集（不相交集类）

堆

算法

广度优先搜索（BFS）

深度优先搜索（DFS）

回溯

插入排序

选择排序

冒泡排序

箱排序

归并排序

快速排序

堆排序

高精度算法

拓扑排序

最小生成树

最短路算法

动态规划

- 链表优化
- 优先队列优化

SPFA 算法

2017 年 6 月
4 日算法讲解

图

二叉树

黑白图像、迷宫

数据结构

队列

栈

并查集（不相交集类）

堆

算法

广度优先搜索（BFS）

深度优先搜索（DFS）

回溯

插入排序

选择排序

冒泡排序

箱排序

归并排序

快速排序

堆排序

高精度算法

拓扑排序

最小生成树

最短路算法

动态规划

- 1 适用范围：
给定的图含有负边权，但是不含有负权环路的图。
- 2 算法思想：
广度优先搜索
某个点进入队列的次数超过 N 次则存在负还。

动态规划

2017 年 6 月
4 日算法讲解

图

二叉树

黑白图像、迷宫

数据结构

队列

栈

并查集（不相交集类）

堆

算法

广度优先搜索（BFS）

深度优先搜索（DFS）

回溯

插入排序

选择排序

冒泡排序

箱排序

归并排序

快速排序

堆排序

高精度算法

拓扑排序

最小生成树

最短路算法

动态规划

- 1 基本概念：
动态规划过程是：每次决策依赖于当前状态，又随即引起状态的转移。一个决策序列就是在变化的状态中产生出来的，所以，这种多阶段最优化决策解决问题的过程就称为动态规划。

基本思想与策略

2017 年 6 月
4 日算法讲解

图

二叉树

黑白图像、迷宫

数据结构

队列

栈

并查集（不相交集类）

堆

算法

广度优先搜索（BFS）

深度优先搜索（DFS）

回溯

插入排序

选择排序

冒泡排序

桶排序

归并排序

快速排序

堆排序

高精度算法

拓扑排序

最小生成树

最短路算法

动态规划

基本思想与分治法类似，也是将待求解的问题分解为若干个子问题（阶段），按顺序求解子阶段，前一子问题的解，为后一子问题的求解提供了有用的信息。在求解任一子问题时，列出各种可能的局部解，通过决策保留那些有可能达到最优的局部解，丢弃其他局部解。依次解决各子问题，最后一个子问题就是初始问题的解。由于动态规划解决的问题多数有重叠子问题这个特点，为减少重复计算，对每一个子问题只解一次，将其不同阶段的不同状态保存在一个二维数组中。

与分治法最大的差别是：适合于用动态规划法求解的问题，经分解后得到的子问题往往不是互相独立的（即下一个子阶段的求解是建立在上一个子阶段的解的基础上，进行进一步的求解）。

适用的情况

2017 年 6 月
4 日算法讲解

图

二叉树

黑白图像、迷宫

数据结构

队列

栈

并查集（不相交集类）

堆

算法

广度优先搜索（BFS）

深度优先搜索（DFS）

回溯

插入排序

选择排序

冒泡排序

桶排序

归并排序

快速排序

堆排序

高精度算法

拓扑排序

最小生成树

最短路算法

动态规划

能采用动态规划求解的问题的一般要具有 3 个性质：

- ① 最优化原理：如果问题的最优解所包含的子问题的解也是最优的，就称该问题具有最优子结构，即满足最优化原理。
- ② 无后效性：即某阶段状态一旦确定，就不受这个状态以后决策的影响。也就是说，某状态以后的过程不会影响以前的状态，只与当前状态有关。
- ③ 有重叠子问题：即子问题之间是不独立的，一个子问题在下一阶段决策中可能被多次使用到。（该性质并不是动态规划适用的必要条件，但是如果没有这条性质，动态规划算法同其他算法相比就不具备优势）

求解的基本步骤

2017 年 6 月
4 日算法讲解

图

二叉树

黑白图像、迷宫

数据结构

队列

栈

并查集 (不相交集类)

堆

算法

广度优先搜索 (BFS)

深度优先搜索 (DFS)

回溯

插入排序

选择排序

冒泡排序

箱排序

归并排序

快速排序

堆排序

高精度算法

拓扑排序

最小生成树

最短路算法

动态规划

动态规划所处理的问题是一个多阶段决策问题，一般由初始状态开始，通过对中间阶段决策的选择，达到结束状态。这些决策形成了一个决策序列，同时确定了完成整个过程的一条活动路线 (通常是求最优的活动路线)。如图所示。动态规划的设计都有着一一定的模式，一般要经历以下几个步骤。

求解的基本步骤

2017 年 6 月
4 日算法讲解

图

二叉树

黑白图像、迷宫

数据结构

队列

栈

并查集（不相交集类）

堆

算法

广度优先搜索（BFS）

深度优先搜索（DFS）

回溯

插入排序

选择排序

冒泡排序

桶排序

归并排序

快速排序

堆排序

高精度算法

拓扑排序

最小生成树

最短路算法

动态规划

- ① **划分阶段**：按照问题的时间或空间特征，把问题分为若干个阶段。在划分阶段时，注意划分后的阶段一定要是有序的或者是可排序的，否则问题就无法求解。
- ② **确定状态和状态变量**：将问题发展到各个阶段时所处于的各种客观情况用不同的状态表示出来。当然，状态的选择要满足无后效性。
- ③ **确定决策并写出状态转移方程**：因为决策和状态转移有着天然的联系，状态转移就是根据上一阶段的状态和决策来导出本阶段的状态。所以如果确定了决策，状态转移方程也就可写出。但事实上常常是反过来做，根据相邻两个阶段的状态之间的关系来确定决策方法和状态转移方程。
- ④ **寻找边界条件**：给出的状态转移方程是一个递推式，需要一个递推的终止条件或边界条件。

数字三角形

2017 年 6 月
4 日算法讲解

图

二叉树

黑白图像、迷宫

数据结构

队列

栈

并查集（不相交集类）

堆

算法

广度优先搜索（BFS）

深度优先搜索（DFS）

回溯

插入排序

选择排序

冒泡排序

桶排序

归并排序

快速排序

堆排序

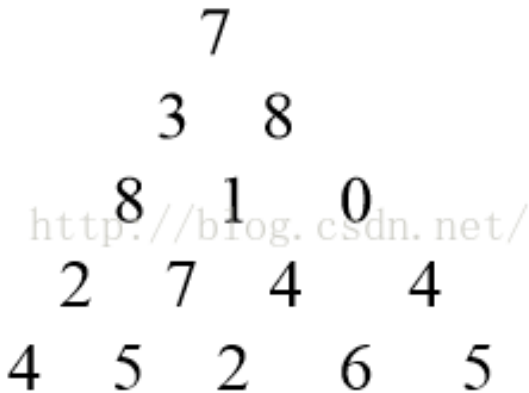
高精度算法

拓扑排序

最小生成树

最短路算法

动态规划



题目

2017 年 6 月
4 日算法讲解

图

二叉树

黑白图像、迷宫

数据结构

队列

栈

并查集（不相交集类）

堆

算法

广度优先搜索（BFS）

深度优先搜索（DFS）

回溯

插入排序

选择排序

冒泡排序

桶排序

归并排序

快速排序

堆排序

高精度算法

拓扑排序

最小生成树

最短路算法

动态规划

- 搜索

HDU 1241 Oil Deposits

HDU 1312 Red and Black

HDU 1026 Ignatius and the Princess I

HDU 1010 Tempter of the Bone

HDU 1242 Rescue

- 回溯

HDU 1627 Krypton Factor

HDU 2553 N 皇后问题

- 并查集

HDU 1213 How Many Tables

题目

2017 年 6 月
4 日算法讲解

图

二叉树

黑白图像、迷宫

数据结构

队列

栈

并查集（不相交集类）

堆

算法

广度优先搜索（BFS）

深度优先搜索（DFS）

回溯

插入排序

选择排序

冒泡排序

桶排序

归并排序

快速排序

堆排序

高精度算法

拓扑排序

最小生成树

最短路算法

动态规划

- 队列、栈、堆、优先队列
HDU 1702 ACboy needs your help again!
HDU 1870 愚人节的礼物
HDU 1387 Team Queue
HDU 1873 看病要排队
- 排序
HDU 1106 排序
- 大数
HDU 1002 A + B Problem II
HDU 1042 N!

参考代码：<https://github.com/xinjiyuan97/NOIP-ACM>

图

二叉树

黑白图像、迷宫

数据结构

队列

栈

并查集（不相交集类）

堆

算法

广度优先搜索（BFS）

深度优先搜索（DFS）

回溯

插入排序

选择排序

冒泡排序

箱排序

归并排序

快速排序

堆排序

高精度算法

拓扑排序

最小生成树

最短路算法

动态规划

- 树
HDU 1710 Binary Tree Traversals
- 最小生成树
HDU 1232 畅通工程
HDU 1233 还是畅通工程
HDU 1863 畅通工程
- 最短路
HDU 2544 最短路
HDU 1847 畅通工程续
HDU 2066 一个人的旅行

2017 年 6 月
4 日算法讲解

图

二叉树

黑白图像、迷宫

数据结构

队列

栈

并查集（不相交集类）

堆

算法

广度优先搜索（BFS）

深度优先搜索（DFS）

回溯

插入排序

选择排序

冒泡排序

箱排序

归并排序

快速排序

堆排序

高精度算法

拓扑排序

最小生成树

最短路算法

动态规划

- 动态规划
 - HDU 1176 免费馅饼
 - HDU 1231 最大连续子序列
 - HDU 1003 Max Sum