

联赛知识点粗略总结

Gromah

August 7, 2015

写在前面

由于本人知识水平及其有限，故只能说是粗略总结，如有错误或者不完整的地方，敬请海涵。

1 基础算法

1.1 模拟

这个好像就只需要一点点代码能力就可以了。

1.2 高精度

这个好像就只需要两点点代码能力就可以了。

1.3 排序

归并排序，快速排序，堆排序 ... 全是 $O(n \log n)$ 的。推荐快排，毕竟 sort 大法好。

1.4 枚举

这个好像没什么好说的。。循环语句会写了就差不多了的样子。

2 搜索

2.1 深度优先搜索

人送外号：dfs。多做点题是王道。
例题：网上一大堆。

2.2 广度优先搜索

人送外号：bfs。多做点题是王道。
例题：网上一大堆。

3 贪心

3.1 最优排列问题

一般都是给你一些事物，不同的安放顺序有不同的权值，问怎么放才最优。
这类问题通常就是考虑两个元素 u, v ，它们如果相邻的话，哪个在前面更优，然后以这个标准来作为两个数的大小比较准则直接排序就可以了。
例题：NOIP 2012 国王游戏。

3.2 最优选择问题

一般就是给你很多个事物，让你选一些事物出来，使得总权值最大或者最小，同时也有一些奇奇怪怪的限制。
这类问题一般都有一个动态规划的解法，但是其复杂度往往不尽人意，所以需要高效的贪心算法来解决问题。
例题：NOIP 2013 花匠。

4 动态规划

人送外号：DP，全称：Dynamic Programming。

通常就是设计一个能表示所有情况的状态，注意要能表示所有情况，然后再来确定转移方程。

4.1 背包问题

一般就是设计一个背包，一般都是一维的，不过也有可能是多维的，因为要表示出所有的情况。

例题：NOIP 2010 乌龟棋

4.2 区间动态规划

一般是要合并一个区间的事物，然后每次可以合并相邻的两个事物，问方案的最大或者最小值。然后就给每个区间设计一个状态，对于每个区间就再枚举断点来转移就可以了。复杂度一般是 $O(n^3)$ 的，有时候也可以优化成 $O(n^2)$ 的。

例题：NOIP 2006 能量项链

4.3 最长单调子序列

一般就是求一个串的最长单调递增子序列。 $O(n^2)$ 暴力不能多说。

有一种 $O(n \log n)$ 的做法。就是维护一个 D_i ，表示在当前所有长为 i 的单调序列中，结尾最小的那个值。然后每次新加入一个元素就二分查找出大于该元素的最小的 D_i ，然后就可以更新 D_i ，也求出了以该元素结尾的最长的单调序列的长度。

其他的最长单调递减子序列，最长单调不增子序列之类的类似。

例题：NOIP 1999 拦截导弹

4.4 最长公共子序列

一般就是求两个串的最长公共子序列。

设计一个二维状态 $Dp[i][j]$ ，表示当前匹配到了第一个串的第 i 位，第二个串的第 j 位，之前的串的最长公共子序列的长度。转移的话分两种情况，如果 $A[i] = B[j]$ ，那么 $Dp[i][j] = Dp[i-1][j-1] + 1$ ，否则就有： $Dp[i][j] = \max(Dp[i][j-1], Dp[i-1][j])$ 。

例题：我都告诉你们怎么做了还要例题吗。。？

4.5 树形动态规划

这个比较灵活，但也比较简单，如果题目做的多了的话自然没什么问题。

例题：codevs 1380 没有上司的舞会。

5 图论

5.1 最短路

多源最短路：Floyd，简单易写。

单源最短路：SPFA，有一个应用就是可以找负环，如果一条边被迭代了很多次那么就有很大的可能存在于一个负环。Dijkstra，有一个堆优化的版本，复杂度稳定，算法核心其实和 SPFA 差不多。

例题：我都告诉你们怎么做了还要例题吗。。？

5.2 拓扑排序

就是给一个有向无环图，让你求一个排列，使得任意一条边的入点都在出点之前出现。这个首先记录所有点的入度，然后找到那些入度为零的点，然后一个一个枚举，并删掉以这个点为入点的所有边，然后就把新出现的入度为零的点加进来，直到所有点都被记录了为止。

当然，也可以用这个来判断图中是否有环。就是如果还存在没有被记录的点，但是这些点中没有入度为零的，那么就说明存在一个环了。

6 字符串

一般都是考 KMP，我在这里就点一下，大家可以自己上网搜一些相关资料。

例题：BZOJ 3620 似乎在梦中见过的样子

7 二分答案

如果是要求满足某个条件的最小值，并且保证如果 i 满足条件，那么对于所有 $j > i$ ， j 都满足条件，那么就可以二分答案了，然后就把问题转化成判定一个答案是否可行了。

大概就是这样的：假设答案一开始一定在 $[l, r]$ 内，我们找到 $mid = \frac{l+r}{2}$ ，然后判断 mid 是否可行，如果可行的话那么最优答案就在 $[l, mid]$ 内，否则就在 $[mid, r]$ 内。二分查找就是这个意思。

例题：NOIP 2012 借教室

8 数论

8.1 快速幂

这个的话感觉不需要多说什么。。。

8.2 素数相关

筛素数的话，有一个 $O(n \log \log n)$ 的算法，大概是这样：从 2 到 n 枚举，如果枚举到的数没有被标记过，那么这个数就是素数，然后枚举这个数的所有倍数并打上标记。

判断素数的话，我们就枚举 2 到 \sqrt{n} 的所有数看有没有一个数能整除 n ，如果没有就说明这个数是个素数。还有一个比较鬼畜的判定方法叫 Miller-Rabin 素数测试，这里不要求掌握，以后才看看吧。

8.3 欧几里得算法

大概就是求两个数的最大公约数。代码简单易写。

还有个拓展欧几里得，是求 $nx + my = \gcd(n, m)$ 的一组整数解的算法，代码类似，应用广泛。

例题：NOIP 2012 同余方程