前三道题目都和PPT内容相适配，第四题为一道杂题。

1. **匹配**(match.c/cpp/pas)

**前40％的数据**：很显然题目就是一个LCS，那么直接用O(n2)的LCS算法即可。

**前100％的数据**：

考虑到一个性质，即A、B两个序列中的元素均**两两不同**。

于是考虑将 A 序列中的元素按1、2、3、4、…、n从左到右依次重新标号，那么每一个原来的标号就对应了一个新的标号。

考虑在B序列中将标号换为对应的新的标号，那么新的B序列上，一个单调上升的序列，对应在 A 序列中下标即单调上升的。也就是说实质上形成了原来A、B序列的一个公共子序列。

于是只用在新的B序列上用O(nlogn)的方法求LIS即可。

1. **块**(block.c/cpp/pas)

**前100％的数据**：

考虑删边后得到的连通块一定是有根的，那么就考虑每个连通块仅在根处统计答案。

令 dp[a][j] 表示以 a 为根的子树，要形成大小为 j 的连通块所需删去的最少边数，注意这个连通块一定包含点 a。另外dp[a][j]中没有考虑点 a 的父亲那边的点，也不需要删除父亲边。

初始值为 dp[a][1]=cnt，其中cnt为点 a 的儿子数目。考虑转移，实质是一个典型的树上背包。现在计算 dp[a][j]，考虑枚举点 a 的每个子结点 b，枚举在 b 这边选大小为k的连通块，则有dp[a][j] = min(dp[a][j-k]+dp[b][k]-1, dp[a][j]) ，其中减1是为了重新连回a和b之间的边。注意和一般的背包相同，为了避免同一物品多次放入，j 应当从大到小枚举。

另外同课上讲的一样，一定注意树上背包时枚举的范围，这样才能做到 O(n2)。细节可以见标程代码。

3 **路径**(path.c/cpp/pas)

**100% 的数据**：

这道题几乎是PPT原题。首先跑一遍tarjan（只用判断是否存在返祖边，不用写完整的tarjan），若有环则输出-1 。

没有环则是一个DAG，考虑使用拓扑排序进行DP即可，dp[i][j]表示以 i 为终点的路径中，字符j 出现的最多次数。

答案即所有dp值取max。

4 **染色**(paint.c/cpp/pas)

**40% 的数据**：

直接用线段树的区间修改即可。

**100% 的数据**：

考虑将染色倒过来，那么一个点被染色之后便将不能再被染色了。此时如果能够O(1)找到一个位置后面第一个没有被染色的点，然后暴力染色，由于每个点只会被染一次，那么复杂度便是O(n)。

问题是如何O(1)找到呢？考虑将所有连续的已经被染色的点（无论染的什么颜色）用并查集合并起来，并且让并查集的根为这段连续区间最右边的点。每次暴力染色时，判断这个点左右是否被染色，如果左边被染过色就将左边的父亲边连向自己，如果右边被染过色就将自己的父亲边指向右边。这样的话，每次通过并查集找到根，再向右走一个位置就是下一个没有被染色过的点。