  ACM结课总结

很幸运这半年接触了acm，让我接触到这么多信院优秀的同學，让我感受数学的强大，[**算法**](http://lib.csdn.net/base/datastructure)的巧妙。又很幸运的在网上看到了大牛一路的成长分享，一遍遍的wa,坎坎坷坷，最终的ac多么大快人心！还记得上学期是怎么过的：疯狂的轮滑，简单的跟老师的节奏让我感到专业知识的空虚。很感谢亲人，与朋友的鼓励与支持，同时还有一个优秀的家人给予我的动力与引导，让我及早认识到学习能力的重要性，让我不甘平凡，坚定的选择了它。

这半年简单的接触，我内心也是跌宕起伏，有过狂热，有过低迷，很疯狂的玩了许久，直到偶然的看到李开复的自传，让我看到了另一种世界，感谢，开复老师的自身经历的分享，让我没有理由去放弃追求自己内心的梦想。

感谢老师提供的机会，让许多人最近都在疯狂的刷acm hdu题，尤其在接触头疼数学题的时候，不断感叹数学的强大，数学的美，让我看到别人分享的代码的时候不断地感叹代码的巧妙！

好拉，简单的总结一下这半年的收获吧：

   对于STL上手还是挺快的，但是要熟练的使用还是有挺多的技巧；基础算法部分的对于主要包括递归递推问题，动态规划问题，广度深度搜索问题，二分,贪心，用于大数运算的高精度计算，感觉这一部分难度提升了很多，需要大量的练习积累和深度的理解。最后一点就是[**数据结构**](http://lib.csdn.net/base/datastructure)，主要是了解了他们的概念，并没有很深的学习和做相关的练习，像栈堆树，图论算法。

Acm一：c++模板库。

很方便的模板库让我们可以简单操作各种c++自带类和函数。比如：排序算法，字符串操作，fifo队列和优先队列，栈，动态数组，关联式容器map容器。

1.排序算法：头文件<algorithm>，sort(begin,end); sort(begin, end, cmp);用它可以暂时不用冒泡法，快排。一个函数可以解决,多么便捷。还有一个生成排序：头文件：<algorithm>，有boolnext\_permutation(begin, end);函数用于改变区间内元素的顺序，产生下一个排列；boolprev\_permutation(begin, en用于产生前一个排列；这样子不用写复杂的递归，多么简单。

2.字符串：头文件<string>,要熟练的巧妙的使用它可以解决很多问题。尤其是自带的reverse函数，但是有些前两天做的hdu上的一个题，关于字符串中每个单词翻转问题，不需要用这个函数，用char数组就可以，或者用栈。 3. 队列：头文件<queue>就是前进先出的特点，优先队列priority会将元素按照优先规则排序。 4.栈：头文件<stack>，先进后出的特点。 5.动态数组：头文件<vector>,各种函数用来操作vector里面的元素，使操作数组更加方便，还有不知道要用内存多大的数组的时候，动态数组就很不错，重要的是它可以存任何类型的元素。 6.关联式容器:头文件<map>,map:这个容器不再限于访问元素是下标为int类型。 .map<下标类型,元素类型>,而且都会根据元素的键值自动排序。Multimap可以两个元素对应相同的键值，但是map不行，m.count(键值)，此键值的元素有几个. 7.Set :头文件<set>,*set和multiset会根据特定的排序准则,或者自己定义排序规则也可以，两者的不同之处在于multiset可以允许元素重复而set不允许元素重复。*

ACM二：基础算法部分

1.对于高精度计算的练习最近在HDU上刚刚应用，原先没有做过相关的练习，也是简单的看了看课本，想要进行大数的运算，熟练的使用也是必须的。高精度加减，很常规了，到了乘法的时候其实是有数组下标的规律的，

c[i+j-1]+=a[i]\*b[i]+x ; x=c[i+j-1]/10; c[i+j-1]%=10;

除法的时候用到了按位相除法，这个相除法也是第一次了解，很巧妙，例如12345/5

1%5=1； 1/5=0;: 每一次的余下的数与下一个数又组成一个十位数，除下的数组起来就是结果，；··········································································

 2. 递推递归： 递推就是找到相邻的数据项间的关系，把复杂的问题一步一步分解成若干步，做递推题的时候，有的典型的递推例如斐波那契数列，很简单，但是有的递推没有接触过要现推到，例如昆虫繁殖的问题，其实仔细推敲都是很容易记住的，还有关于图形切割的递推，没有一个不在看你的数学能力。这部分题刚看起来都让人不知所措，尤其感觉我的数学底子很一般，所以需要不断地积累各种规律公式，以前在数学上挖的坑还是要慢慢填上的。

有些题你是没有见到过的，这就要看你的数学推理了。特别典型的题目就是汉诺塔问题，

关于汉诺塔，搞清一小步是怎样走的就行，但是挺绕有没有。

3.然后就是动态规划：动态规划可以将你不可想象的那些特别复杂的数学问题分步给求出去最优解，引入了一个名词，状态变量，每一步都是联系上一步的状态寻求这一步的状态值，然后一步一步到求出每一个的状态，并且以每一步最优的状态来保证全局最优的状态。最典型的就是三角形，求出从顶到底元素数值最大的路径的和是多少？第一眼看到这类题，我的天，这么复杂的问题。。。。但是了解动态规划的状态变量就会发现，复杂的问题其实可以简单化、这个如果从上往下不知道下一层哪一个路径更优，所以可以从倒数第二层开始，求出每一个元素到最后一层的最大的状态值，然后以此类推到第一层。

关于动态规划还有最长上升子序列，最长子序列和，最长m子序列和问题，以及最长上升子序列的和最长子序列和的二维形式，分别做过滑雪问题，和最大子矩阵和问题。今天做hdu做这部分题的时候发现生疏了~~~还有就是背包问题了，有三种背包问题，分别是01背包，完全背包和多重背包。记得刚开始理解的时候很难理解有木有，但是每一种背包都有一种模板：

0-1背包模板：for i=1…..n; for v=v……0; f[v]=max{f[v],f[v-c[i]]+w[i]};体积是从大到小。完全背包模板：for=1……n;for  v=0…..v;f[v]=max{f[v],f[v-cost]+weight};体积为从小到大，

这类题都是特别的典型，一眼就能看出来是什么背包，这类题型的思维比较固定。

4.贪心：贪心不是从整体上考虑，而是最优局部解上考虑，最后这个局部解就是最优解，这个部分可以和背包问题联系，如果物体不可以分割那就得用动态规划，如果可以分割用贪心算法，做一个题的时候贪心算法不容易被人想到。

5.二分三分：这部分算法理解起来比较容易，并且在查找元素的时候可以提高很高的效率。并且扩展出去知道问题解得范围，并且有单调性，可以不断地二分验证求出答案。最典型的就是切绳子问题，n条切成k段，最长的长度是多少。思路就是是在某一区间内查找元素，如果查找到了元素，结束；如果未查找到，目标元素在左边，更改上界，目标元素在右边，更改下界，知道找到或者逼近为止。对于二分三分的总结就是：对于一些实际问题，当公式难以推导出来时，二分、三分法可以较为精确地求解出一些临界值，例如一些函数的极大值，极小值。

6.搜索：有深度搜索和广度搜索。深度搜索就是找到一条路一直走下去看看符不符合条件，如果不符合的话，回溯，继续向下找另一条路；广度搜索就是每一个层级的可能性都给列出来，然后一层一层的向下找，如果找到就停止。这种算法就是有方向性的遍历，一直到找到目标。广度优先搜索用队列进行处理。深度优先搜索相当于递推问题，重在回溯。一看看这个算法的时候。上课一些一看上去特别复杂的题目，用这个算法可以轻松得出结果，对于

刚开始接触这一算法概念的时候，心里默默感叹这一算法的强大。虽然事后知道要枚举，效率可能并不是很高。当然自己在看例题的时候，看起来也是交瘁，每一种搜索都有模板，但是在回溯的时候要恢复状态继续深搜在理解起来的时候很费劲有木有，在做专题的时候深深感受到搜索的万般变化,这个专题开了一个月~~，然而自己就做了7道题有没有，到现在搜索做的并不好，感觉自己学的很浅，每一种算法的学习和使用都要慢慢的做过入门题，慢慢体味，用入门题帮助重新认识这一个算法。

ACM三：数据结构

1.对于栈队在ST,L学习怎么使用现成的模板，而这一部分学习的是这些模板是怎样写出来的，他的原理，实现的方法，栈是先进后出，队列是先进先出，都是用指针控制。自己对于这方面的知识也只是了解。

2.还有非线性的数据结构的树类型，对于我这个小白的感觉就是树很强大，在计算机广泛的应用，数据结构是[**数据库**](http://lib.csdn.net/base/mysql)层次的基础，也是各种索引和目录的主要形式，而且也是各类比赛中的必备知识，前面说到搜索的效率可能不会那么高，那么搜索在树形结构的基础上搜索效率会提升很多。树的存储结构有四种方法，父亲表示法，孩子表示法，父亲孩子表示法，孩子兄弟表示法，各有各的特点；还有树的遍历也有很多方法，常用的有先序遍历，后序遍历，层次遍历。对于树的特殊结构二叉树：它就有一些性质，二叉树的存储结构：链式存储结构和顺序存储结构；它的遍历方法：访问根节点，先序遍历左子树，先序遍历右子树；此外还有二叉树的其他操作例如建立一颗二叉树，删除二叉树，插入一个结点到二叉树，在排序二叉树中查找一棵树，找到返回该结点，；

还有一种结构叫做堆，堆结构是一种数组对象，可以视为一种完全二叉树，每一结点的子树的值和结点的值的关系有可以分为大根堆和小根堆。怎么操作堆，主要有两个操作，往堆中加一个元素和从对中提取一个元素。在这里可以用堆排序来进行排序。

4.最后一个专题就是图论算法，存储结构有两种方法，第一种是二维数组邻接矩阵存储，但是不适合特别大的图，会浪费很多的存储空间；第二种方法是数组模拟邻接表存储，这个又叫链式存储。图的遍历有深度优先遍历和广度优先遍历，这个和深搜，广搜相似，在图的遍历里，广度优先遍历用的很少。对于图的遍历有两个大的问题就是，一笔画问题，和哈密顿环问题。对于图论最短路径问题有几种常用的算法:bellman-ford算法，spfa算法，是bellman-ford的改进，dijkstra算法,floyd-washall（弗洛伊德）算法。最后这两章学的有些匆忙，并没有仔细体味。

最后的最后是关于数学方面的问题，log的使用啊，菲波那切数列的公式方法啊，卡斯兰特数啊，筛素数啊，在做hdu时候就感觉脑壳疼，题目给的数大的吓人，不用数学方法做百分百超时，数学的神坑还是要一点一点填上的。就像跟室友一块做题，感受到数学不好被虐是什么感受，不说话，静静填坑。数学与算法的关系是不言而喻的，要想在算法上有所收获，还是要静下心学好数学。

谈谈总结：

这半年的成长是无形的，在做各种题的时候也是在不断积累的经验和知识时候，这半年收获的不仅仅是知识，还有锻炼了自己的思维能力和代码的实现能力。这半年的成长，仅仅也是小白菜的开始，还记得去年参加的acm新生邀请赛，一个简单的爬楼问题，就是简单的裴波那契数列都不会啊，现在的水平就是会了而已，这半年只是了解acm的大概，嗯仍然需要学习积累像更多的斐波那契数列的那样想当年不会的问题，跟需要更深度的理解各类算法。

成长路上：再想想这半年的学习过程也是也想做一道题一样wa,wa,然后ac,中途也有不少的诱惑，也有不少做题时的wa时候的憔悴，憔悴过后收获了什么，收获了碰到相同问题的时候直接ac的成就感，收获了CSDN上别人的分享：让人钦佩的代码，写代码的小技巧，还有他们一路成长的心路历程。当你了解到他们仍在疯狂的刷oj上的sratus的时候，废寝忘食的敲着键盘的时候，让我心生敬佩，内心多了些感慨。感谢这半年一路上室友的陪伴和同学的陪伴，这俩人都省赛铜奖让人情何以堪~最后感谢老师，这半年的辛付出和鼓舞，鼓励同学都能在acm路上走更远，希望山农的acm在老师的带领下,刷新历史。

今天：现在想想 hdu steps100道题多么刺激，从头来一遍知识的梳理，又是多么的舒畅。所以现在，还是珍惜机会，我希望这份总结不是结束，而是真正意义上的开始。2017，我们依旧渺小。