程序设计概述

ACM

主要内容

ACM / ICPC简介

题目示例

相关知识

相关网络资源

ACM / ICPC简介

由国际计算机界历史最悠久、最具权威性的组织ACM学会（Association for Computer Machinery）主办。目前，ACM国际大学生程序设计大赛已经成为参赛选手展示计算机才华的广阔舞台，是著名大学计算机教育成果的直接体现，也是信息企业与世界顶尖计算机人才对话的最好机会，因此，该赛事已逐步演变成一项全球高校间计算机学科实力的竞赛。

ACM

ACM / ICPC简介

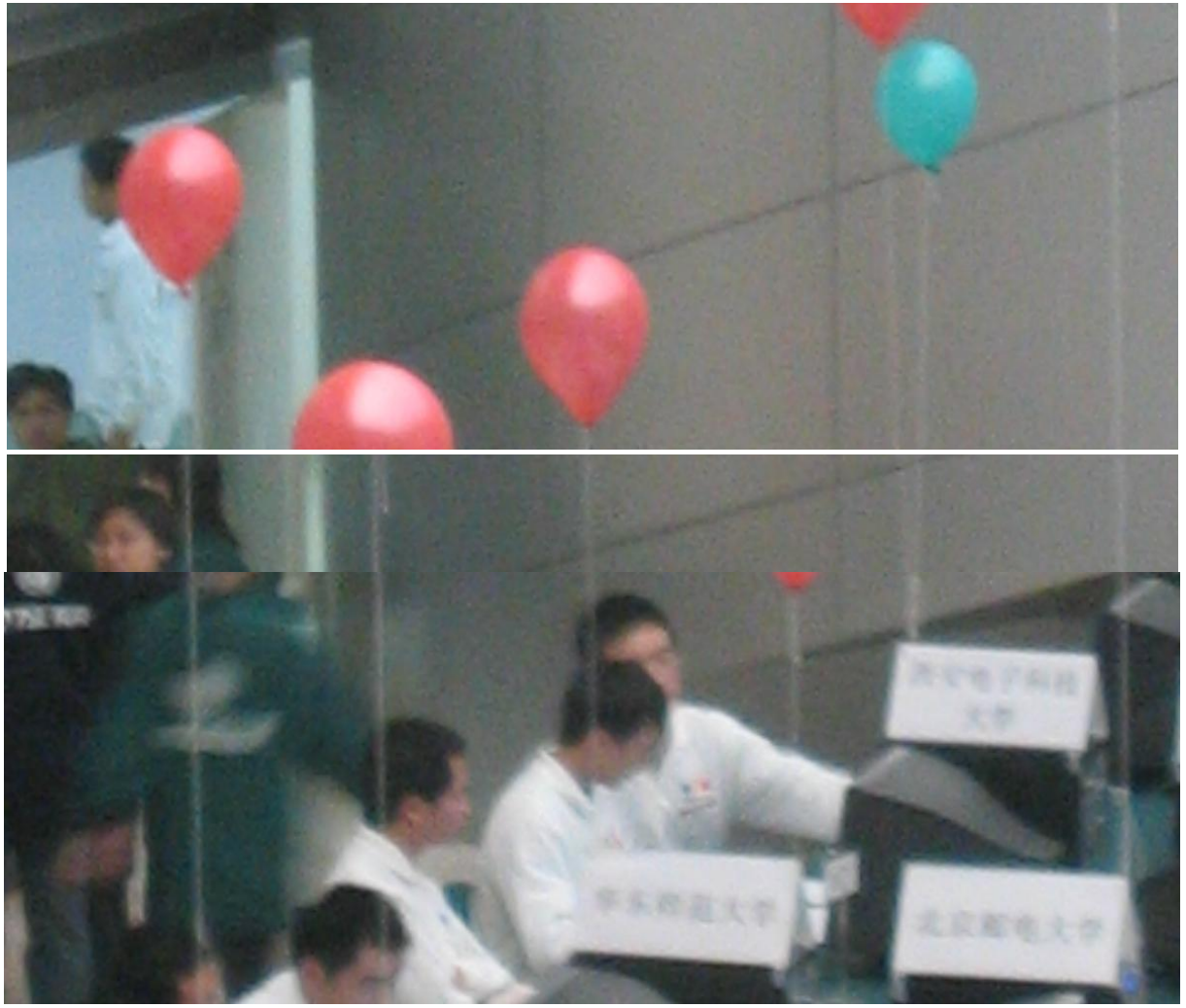
ACM / ICPC竞赛题目难度大，更强调算法的效率，竞赛时 3人合作，共用一台计算机，非常注重团队协作精神；很多题目没有成熟的算法，旨在考验参赛队员的创新能力；比赛现场完全封闭，现场提交程序、现场评判，能客观、公正地展现参赛学生的真实水平。

ACM

ACM / ICPC简介

ACM

ACM / ICPC简介



ACM / ICPC简介

常用编程语言

- C/C++
- Java

重点：语言的标准库

ACM

Agenda

ACM / ICPC简介

题目示例

相关知识

相关网络资源

ACM

Machine Schedule

Problem G: Machine Schedule

(input file: machine.in)

As we all know, machine scheduling is a very classical problem in computer science and has been studied for a very long history. Scheduling problems differ widely in the nature of the constraints that must be satisfied and the type of schedule desired. Here we consider a 2-machine scheduling problem.

There are two machines A and B. Machine A has n kinds of working modes, which is called mode_0, mode_1, ..., mode_n-1, likewise machine B has m kinds of working modes, mode_0, mode_1, ..., mode_m-1. At the beginning they are both work at mode_0.

ACM

Machine Schedule(cont.)

For k jobs given, each of them can be processed in either one of the two machines in particular mode. For example, job 0 can either be processed in machine A at mode_3 or in machine B at mode_4, job 1 can either be processed in machine A at mode_2 or in machine B at mode_4, and so on. Thus, for job i, the constraint can be represent as a triple (i, x, y), which means it can be processed either in machine A at mode_x, or in machine B at mode_y.

Obviously, to accomplish all the jobs, we need to change the machine's working mode from time to time, but unfortunately, the machine's working mode can only be changed by restarting it manually. By changing the sequence of the jobs and assigning each job to a suitable machine, please write a program to minimize the times of restarting machines.

ACM

Machine Schedule(cont.)

Input

The input file for this program consists of several configurations. The first line of one configuration contains three positive integers: n,m (n,m < 100) and k (k < 1000). The
the k jobs, each
The input w ill be t
zero.

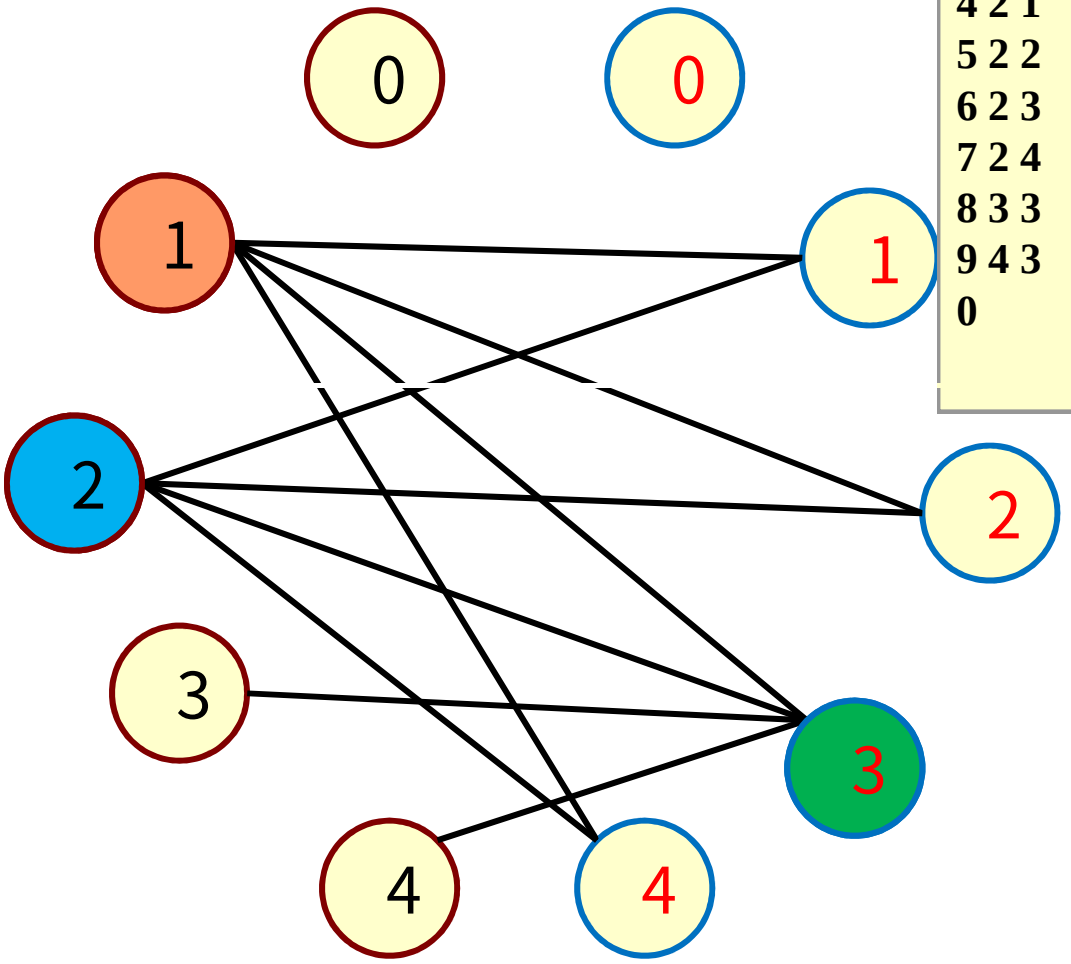
Output

The output shoud l
m inim al tim es of

Sample input	Sample output
5 5 10 0 1 1 1 1 2 2 1 3 3 1 4 4 2 1 5 2 2 6 2 3 7 2 4 8 3 3 9 4 3 0	3

ACM

解题思路




```
5 5 10
0 1 1
1 1 2
2 1 3
3 1 4
4 2 1
5 2 2
6 2 3
7 2 4
8 3 3
9 4 3
0
```

以任务作为边，机器的模式为点。如果一任务可以由 A机器的模式 i或 B机器的模式 j加工，那么 i,j连边，边代表该任务。于是问题变成了求以最少的点来覆盖所有边的问题，即最小点覆盖，即可在转变成由求最大匹配得到。运用著名的匈牙利算法即可快速的求解该问题。需要注意的是由于机器开始都处于 0模式，所以如果有任务可以有任一种机器的 0模式加工的话可先直接加工该任务而不要重启机器，即不要连边。

ACM

典型题例：最大子段和

问题：给定 n 个整数（可能但不全为负）
$a_1, a_2, \ldots, a_n$，求 i, j 使 a_i 到 a_j 的和最大。

例如：6 个数：(-2, 11, -4, 13, -5, -2)

最大和记做 $ms(p, q)$，表示 a_p 到 a_q 这一段的最大和。显然 $ms(1, 6) = 20$, i = 2, j = 4。

ACM

最大子段和解法

最简单的是 3重循环： 2重遍历， 1重求和。时间复杂度： $O(n^3)$

- 改进方法之一：可以通过 sum(i:j-1)+ a(j)来求 sum(i:j)，代码见下页。时间复杂度： $O(n^2)$
- 改进方法之二：先计算 psub(k)= sum(1:k)，然后 sum(i,j)= psub(j)-psub(i-1)，代码略。时间复杂度 $O(n^2)$
- 改进方法之三：动态规划法！

ACM

典型题例：最大子段和

```
intMaxSum(intn,inta[],int&besti,int&bestj)
{
    intsum=0;
    for(inti=1;i<=n;++i)
    {
        intthissum=0;
        for(intj=i;j<=n;++j)
        {
            thissum+=a[j];
            if(thissum>sum)
            {
                sum=thissum;
                besti=i,bestj=j;
            }
        }
    }
    returnsum;
}
```

ACM

最大子段和的DP解法

记b[j]=max{sum(i:j) ,1≤i≤j} ,则max{b[k],1 ≤ k≤ n}就是ms(1 ,n), 即所求的全局最大和。

b[j]=max(b[j-1]+ a[j],a[j]),1<= j<=n

根据上式容易设计出最大子段和的DP算法，复杂度仅为O(n)，详见代码。

ACM

最大子段和的DP解法

```
intMaxSum(intn, inta[])
{
    int sum = 0, b = 0;
    for(int i=1; i<= n; ++ i)
    {
        if ( b > 0 )
            b += a[i];
        else
            b = a[i];

        if ( b > sum )
            sum = b;
    }
    return sum;
}
```

ACM

Agenda

ACM / ICPC简介

题目示例

相关知识

相关网络资源

ACM

相关知识

基本数据结构

- 线性结构
  - 数组
  - 链表
  - 栈和队列
  - **Hash**表
  - 串
- 非线性结构
  - 树（堆、二叉树等）
  - 图

ACM

相关知识

基本算法设计策略

- 分治法
- 贪心法
- 动态规划
- 回溯法
- 分枝－限界法

时间、空间复杂度分析方法

ACM

相关知识

图论相关知识
组合数学知识
计算几何知识
数论知识
程序设计语言
其它（博弈、运筹学...）

ACM

Beginner's Guide

根据自己的实际情况补充 C++（类的基本概念、语法、开发环境的基本使用、STL）、数据结构的基本知识。

听讲座：基本数据结构、基本算法（分治、贪心、DP、搜索）。

结合讲座内容做一些题目。

看书——《实用算法的分析与程序设计》。

做 usaco 上的 step by step。

参考 oibh 上《入门 FAQ》

ACM

Agenda

ACM / ICPC简介

题目示例

相关知识

相关网络资源

ACM

相关网络资源

UsacoGate：http://ace.delos.com/usacogate

UsacoContest：http://ace.delos.com/contestgate

USACO：http://www.usaco.org/

UsacoGate是全美计算机奥林匹克竞赛（USACO）的一个训练网站，要参加 USACO 就必须参加 UsacoGate的训练。每年都有 USACO Spring，Fall，Winter，以及 WinterCamp的全美竞赛，并且提供网络竞赛。

UsacoGate的难度由浅入深，分了若干个专题，每个专题都有一篇讲座以及４～５道题目供练习（提供在线即时评测系统）。个人认为，上面的有些题目比较经典，也有些题目不好，总的来说还是值得已经熟悉程序设计语言但是对算法和数据结构不太了解的学生去做的。每道题目后面都有解答，附有 C++ 程序，建议初学者多看看别人的解答和程序。

http://wzjoiwzms.com/usaco/default.asp上有USACO上的题目和Text的翻译。

ACM

相关网络资源

http://acm.zju.edu.cn/

浙大

http://acm.pku.edu.cn/JudgeOnline

北大

http://icpc.baylor.edu

比赛官方网址