## 大题类型：

1.结合数据结构设计算法（exam1之前的内容）
2.DP
3.network flow
4.prove NPC
5.转化linear programing  ok

复习计划：至少完成倒数8套卷子题型训练
11.22：exam1知识点复习，题型1
11.23：exam1知识点复习，题型1
11.24：LP知识点总结，题型5
11.25：LP知识点总结，题型5
11.26：NPC知识点总结，题型4
11.27：NPC知识点总结，题型4
11.28：DP知识点复习，题型2
11.29：NW知识点复习，题型3
11.30：整合查漏补缺

白天自行复习：笔记和之前的考试总结，提前看晚上要总结的题目
晚上：讨论或者自己总结

Part1：

1.如何在0（n）建堆
2.关于二项堆：
  一个二项堆有O（logN）颗二项树
3.关于fibonacci heap：
  不得不做的时候再调整堆的结构
  斐波那契堆(Fibonacci heap)是堆中一种，它和二项堆一样，也是一种可合并堆；可 用于实现合并优先队列。斐波那契堆比二项堆具有更好的平摊分析性能，它的合并操作的时间复杂度是O(1)。
与二项堆一样，它也是由一组堆最小有序树组成，并且是一种可合并堆。
与二项堆不同的是，斐波那契堆中的树不一定是二项树；而且二项堆中的树是有序排列的，但是斐波那契堆中的树都是有根而无序的
4.divide and conquer 中子问题规模必须一样大么？
5.dijkstra，bell-man ford 有向和无向图都能run

Greedy 两种证明题型总结：
15 spring ：step head 思路-----类似schedule问题的解决interval问题，采用这种思路

4) 16 pts
We've been put in charge of a phone hotline. We need to make sure that it's staffed by at least one volunteer at all times. Suppose we need to design a schedule that makes sure the hotline is staffed in the time interval $[0, h]$. Each volunteer $i$ gives us an interval $[s_i, f_i]$ during which he or she is willing to work. We'd like to design an algorithm which determines the minimum number of volunteers needed to keep the hotline running. Design an efficient greedy algorithm for this problem that runs in time $O(n \log n)$ if there are $n$ student volunteers. Prove that your algorithm is correct.
You may assume that any time instance has at least one student who is willing to work for that time.

证明：1）假设有一个最优算法，然后通过inductive 思想来比较我们设计的greedy和这个最优算法

2）basic case：根据我们选择的要求，来说明我们选择的solution至少和按最优算法的solution一样好

3）induction hypothesis：对于第r个，我们用我们的greedy 算法挑选的解来代替最优算法的解，同样也是一个最优算法。

4）所以我们的算法也是一个最优算法，并且solution 的size一样大

15summer：exchange variable思路------当只涉及到顺序排序结果作为解的时候考虑这种思路

2) 16 pts
You are given n jobs of known duration $t_1, t_2, \dots, t_n$ for execution on a single processor. All the jobs are given to you at the start and they can be executed in any order, one job at a time. We want to find a schedule that minimizes **the total time spent by all the jobs** in this system. The **time spent** by one job is the **sum of the time spent on waiting plus the time spent on its execution**. In other words, the total time spent by all jobs is the total sum of their finish times. Give an efficient solution for this problem. Analyze the complexity of your solution and prove that it is optimal.

证明：1）Assume 存在一个最优算法存在至少一个inversion

2）我们exchange这个inversion，比较exchange前后的变化

3）发现exchange后我们设计的greedy 算法比这个算法至少更好

4) exchange all inversion, 得到的就是我们设计的greedy 算法

Part2：

LP总结：

题型一：standard form---待定

题型二：integer program

变量和限制条件都是discrete的，比如一个萝卜一个坑的特征

15spring

6) 16 pts

There are n people and n jobs. You are given a cost matrix, C, where C[i][j] represents the cost of assigning person i to do job j. You want to assign all the jobs to people and also only one job to a person. You also need to minimize the total cost of your assignment. Can this problem be formulated as a linear program? If yes, give the linear programming formulation. If no, describe why it cannot be formulated as an LP and show how it can be reduced to an integer program.

Solution:

We need a 0,1 decision variable to solve the problem and therefore we need to formulate this as an integer program. Below is a formulation of integer program.

Let $x_{ij}$ = 1, if job j is assigned to worker i.

= 0, if job j is not assigned to worker i.

Objective function: Minimize

$$\sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij} x_{ij}$$

Constraints:

$$\sum_{i=1}^{n} x_{ij} = 1, \quad for\ j = 1,2,\ldots,n$$

$$\sum_{i=1}^{m} x_{ij} = 1, \quad for\ i = 1,2,\ldots,m$$

$$x_{ij} = 0\ or\ 1$$

一个萝卜一个坑的特征表现在：

每一个工作只能一个人来做

每一个人只能做一份工作

题型二：讲一个问题reduce成一个LP问题

**1）先考虑varibale是什么**

**2）考虑linear constraint是什么**

3）objective function是什么？

15summer：

6) 16 pts

A company makes three products and has 4 available manufacturing plants. The production time (in minutes) per unit produced varies from plant to plant as shown below:

| | | Manufacturing Plant | | | |
|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 |
| Product | 1 | 5 | 7 | 4 | 10 |
| | 2 | 6 | 12 | 8 | 15 |
| | 3 | 13 | 14 | 9 | 17 |

Similarly the profit ($) contribution per unit varies from plant to plant as below:

| | | Manufacturing Plant | | | |
|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 |
| Product | 1 | 10 | 8 | 6 | 9 |
| | 2 | 18 | 20 | 15 | 17 |
| | 3 | 15 | 16 | 13 | 17 |

If, one week, there are 35 working hours available at each manufacturing plant how much of each product should be produced given that we need at least 100 units of product 1, 150 units of product 2 and 100 units of product 3. Formulate this problem as a linear program. You do not have to solve the resulting LP.

变量 ：不要忽略Xij这种形式的变量设计

xij = amount of product i (i=1,2,3) made at plant j (j=1,2,3,4) per week.

Constrain：仔细读题，不要漏

We first formulate each constraint in words

- Limit on the number of minutes availab

5x11 + 6x21 + 13x31 <= 35(60)

7x12 + 12x22 + 14x32 <= 35(60)

4x13 + 8x23 + 9x33 <= 35(60)

10x14 + 15x24 + 17x34 <= 35(60)

- Lower limit on the total amount of each

x11 + x12 + x13 + x14 >= 100

x21 + x22 + x23 + x24 >= 150

x31 + x32 + x33 + x34 >= 100

All variables are greater than equal to zero.

objective：

Maximize

10x11 + 8x12 + 6x13 + 9x14 + 18x21 + 20x22 + 15x23 + 17x24 + 15x31 + 16x32 + 13x33 + 17x34

NPC题型总结 ：X<p Y

考虑如何构建Y问题，使得当我询问得到Y 问题的solution的时候，就能保证我这个solution可以以一种机制对应到X的solution。

那么我如何完成这种构建呢？从上面可以看出，我要根据X的样子，同时满足两个条件构造出Y样子的问题。两个条件是（1）构建出来的问题，是我Y的这种形式（2）构建出来的问题一旦有solution，那我就能根据这个solution对应到X的一个solution

1)SAT和图的reduction：切记逻辑的非黑即白的原则的在构建图的应用，比如通过增加dummy variable进而对整个clause进行修改（长度变化）

2）图和图的reduction: 一般考虑怎么增加边，增加点，使得X问题能以Y的形式表达出来

3）问题本身的reduction：首先考虑Y问题在什么情况下，就等同于X问题（也就是等于它本身），因为这种情况本身就是已知的NPC问题了，所以其他情况（问题Y不等同于X的时候），要想办法往这种情况reduce。
Ex. 15 spring ---independent set <p HALF IS