东 南 大 学 考 试 卷（A 卷）

课 程 名 称	算法设计与分析	考 试 学 期	2018-2019-2	得分	
适 用 专 业	计算机	考 试 形 式	开卷	考试时间长度	150 分钟

（可携带纸质教材、课件、讲义、笔记）

1. 判断题（共 10 分，每小题 2 分）

a）贪心算法是最优求解算法..(X)

b）KMP 算法为长度为m的串计算 next 数组的复杂度为$O(\log m)$..............................(X)

c）已知一个 NP 完全问题 A 和一个 NP 问题 B，若 A 可多项式时间规约到 B 问题的一个实例，则 B 也可多项式时间规约到 A 的一个实例..(√)

d）算法 A 的时间复杂度为$T_A = O(n\log n)$，则$T_A = O(n^2)$也正确 ....(√)

e）动态规划算法可以多项式时间求解...(X)

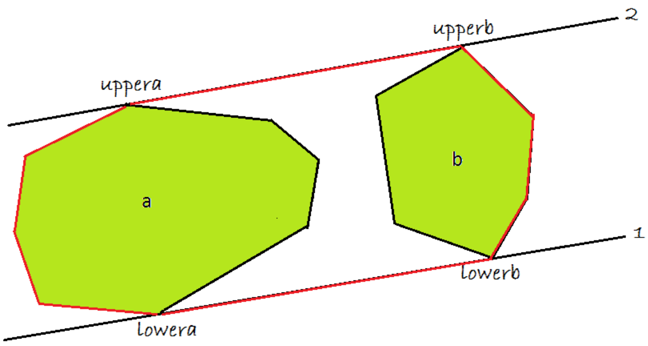
2. 给定平面上n个坐标点$P = \{p_1, p_2, \ldots, p_n\}$，请完成以下方法构建凸包。（共 10 分）

a）如何将这n个点均分成两部分，使得两部分坐标点数相差最多为 1 个？（4 分）

b）如何合并两部分结果得到最终结果？（5 分）

c）算法的时间代价主要包含哪些部分？时间复杂度为多少？（1 分）

参考答案：已知一个点集合，我们已经知道了它们的凸包。假设我们已经知道了左半部分和右半部分的凸包，那么现在的问题就变成了合并这两个凸包，并且确定整个集合的凸包。这一步可以通过找到左右两个凸包的上下切线来搞定。这就是两个凸多边形之间的切线。设左边的凸包为 a，右边的凸包为 b。于是下面的切线和上面的切线分别为 1 和 2，如图所示。然后红线画出了最后的凸包。

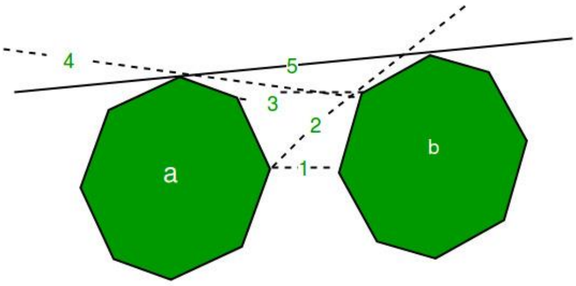


现在问题还没完全解决，我们如何才能找到左右整个的凸包呢？这里递归在图中出现了，我们可以把集合中的点分区，直到每个集合中的点的数目都非常小，例如 5 个点，然后我们就可以通过暴力法找到这个集合的凸包了。对于整个集合的点，在合并的过程中就可以得到整个凸包。

For finding the upper tangent, we start by taking two points. The rightmost point of a and leftmost point of b. The line joining them is labelled as 1. As this line passes through the polygon b (is not above polygon b) so we take the anti-clockwise next point on b, the line is labelled 2. Now the line is above the polygon b, fine! But the line is crossing the polygon a, so we move to the clockwise next point, labelled as 3 in the picture. This again crossing the polygon a so we move to line 4. This line is crossing b so we move to line 5.

Now this line is crossing neither of the points. So this is the upper tangent for the given polygons.

For finding the lower tangent we need to move inversely through the polygons i.e. if the line is crossing the polygon b we move to clockwise next and to anti-clockwise next if the line is crossing the polygon a.



3. 给定n个整数$a = \{a_1, a_2, \dots, a_n\}$，以及一个整数$B$。要求在$n$个整数中找一子集，使得这些子集中整数和刚好等于$B$。请设计动态规划算法求解该问题。（共 10 分）

a） 请写出伪多项式时间复杂度的动态规划迭代公式。（6 分）

b） 请特别讨论初始化条件。（4 分）

参考答案：

$$f(i,b) = \begin{cases} \max\{f(i-1, b-a_i), f(i-1,b)\} & if\ b \geq a_i \\ f(i-1,b) & if\ b < a_i \end{cases}$$

初始条件：

$$f(0,b) = 0$$
$$f(i,0) = 1$$

4. 给定一无向图 $G = (V, E)$，每条边$e \in E$有一非负权值$\omega(e)$，每个顶点$v \in V$有一非负权值$\omega(v)$，一条顶点s到顶点t的路径上所有顶点和边的权值和是这条路径的长度。请设计方法求顶点s到顶点t的最短路径长度。（共 10 分）

参考答案：令$d(t)$表示s到t的最短路径长度，$d(t) = \min\limits_{v \in N(t)} \{d(v) + \omega(vt) + \omega(t)\}$，其中 $\omega(\cdot)$表示顶点或边的权值，$N(t)$表示t的邻居顶点。

5. 给定一个数组 $T[1,2,\dots,n]$，存储了第 1 天到第 n 天的平均气温。假设气温变化没有规律。现要求为每一天 i 计算一个 $Warmer[i]$，记录后续气温中第一个更高气温的日期。

$Warmer[i] = \min\{k \mid 1 \leq k \leq n, T[k] > T[i] \text{ and } k > i\}$.

如果不存在，则设置 $Warmer[i] = n+1$。例如下面是一个共 7 天的例子。（共 10 分）

i =	1	2	3	4	5	6	7
$T[i]$=	33	24	24	32	35	29	36
$Warmer[i]$=	5	4	4	5	7	7	8

（a）请设计一算法计算数组 $Warmer[1,2,\dots,n]$。

（b）请设计一时间复杂度为$O(n)$的算法求解该问题。

参考答案：

Warmer-day ($T[1..n]$)　　//output $Warmer[i]$

1　$S \leftarrow \varnothing$　　　//S is a stack.

2　$T[0] \leftarrow \infty$　　　//dummy element

3　$T[n+1] \leftarrow \infty$　　　//dummy element

```
Push(S, T[0])
for k ← 1 to n+1
      while Top(S) < T[k]
            T[i] ← Pop(S)     //T[i] is the top element
            Warmer[i] ← k
      endwhile              //T[0] = ∞ will stay at the bottom of the stack
      Push(S, T[k])               //Warmer[k] is not known yet for elements in stack
 endfor
return Warmer[1..n]
End
```

6. 若数组中某一元素出现次数超过一半，则称该元素为 majority 元素。给定一个整型数组$a[1,2,\dots,n]$，请按要求设计方法寻找 majority 元素。（共 10 分）

a）请设计一时间复杂度为$O(n\log n)$的方法寻找 majority 元素。（6 分）

b）请设计一时间复杂度为$O(n)$的方法寻找 majority 元素。（4 分）

参考答案：

a）将一个数组分为左右半区，如果一个元素是整个数组的主要元素，那么它一定也是左半区或者右半区的主要元素，不然个数不可能超过 n/2。假设我们已经得到了左右半区的主要元素（如果哪个半区没有主要元素，则返回值为-1），下面就是如何合并。我们将左右半区合起来，数出左半区主要元素或者右半区主要元素出现的次数是否超过这整个区的一半，谁超过就返回谁。如果都没有超过，就返回-1。

```
int get_majority_element(vector<int> &a, int left, int right) {
   if (left == right) return -1;      //没有元素
   if (left + 1 == right) return a[left];    //如果只有一个元素
   int mid = (left + right) / 2;
   int low = get_majority_element(a, left, mid);
   int high = get_majority_element(a, mid, right);
   int m = 0, n = 0;
   for(int i = left; i < right; i++){
      if(a[i] == low) m++;
      if(a[i] == high) n++;
   }
   if(m > (right - left)/2) return low;    //统计次数
   if(n > (right - left)/2) return high;
   return -1;
}
```

b）

```
res=a[0], count=1;
for(int i=1; i<n; i++) {
   if(a[i]==res) {count++;}
   else{count--;}
   if(count<=0) {
```

```
        res = a[i];
        count=1;
    }
}
count=0;
for(int i=0; i<n; i++) {
    if(res==a[i]) {
        count++;
    }
}
if(count>n/2)print("find the majority number");
else print("no majority number");
```

7. 骨牌排放搜索问题。（共 10 分）

一副骨牌共 28 张，由以下点数组合构成：

骨牌号	点数	骨牌号	点数	骨牌号	点数	骨牌号	点数
1	0 | 0	8	1 | 1	15	2 | 3	22	3 | 6
2	0 | 1	9	1 | 2	16	2 | 4	23	4 | 4
3	0 | 2	10	1 | 3	17	2 | 5	24	4 | 5
4	0 | 3	11	1 | 4	18	2 | 6	25	4 | 6
5	0 | 4	12	1 | 5	19	3 | 3	26	5 | 5
6	0 | 5	13	1 | 6	20	3 | 4	27	5 | 6
7	0 | 6	14	2 | 2	21	3 | 5	28	6 | 6

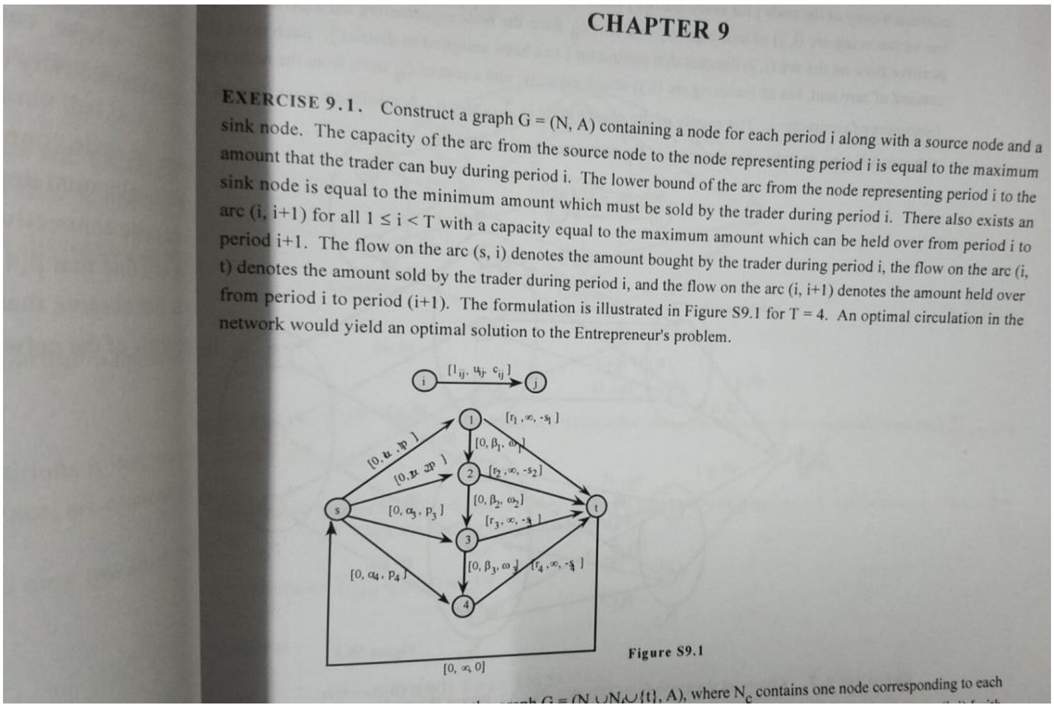
这 28 张牌可以拼成 7×8 的点数网格，例如：

点数网格	骨牌号图
6 6 2 6 5 2 4 1	28 28 14 7 17 17 11 11
1 3 2 0 1 0 3 4	10 10 14 7 2 2 21 23
1 3 2 4 6 6 5 4	8 4 16 25 25 13 21 23
1 0 4 3 2 1 1 2	8 4 16 15 15 13 9 9
5 1 3 6 0 4 5 5	12 12 22 22 5 5 26 26
5 5 4 0 2 6 0 3	27 24 24 3 3 18 1 19
6 0 5 3 4 2 0 3	27 6 6 20 20 18 1 19

使用搜索算法从点数网格求对应的骨牌号图，描述状态空间的组织，有可能的话，给出相关剪枝策略。

参考答案：解法很多，可以按位置组织搜索树，也可以按骨牌号组织搜索树，还可以用舞蹈链求解，56 个位置和 28 张牌组成 84 列，共有 7×7+6×8 行，每行在相邻两个位置和对应骨牌号填 1，其余填 0。

8. 一家专卖店连续N天营业，只经营一种商品，第i天可以最多进货a_i台，进货单价p_i，可以最多将b_i台存放在仓库，单位存放费用为ω_i，必须为已签合同的客户供货c_i台，单价为s_i，以该价格该天能卖到断货。请构建一个费用网络流模型，计算利润最大的营销方案。（共 10 分）

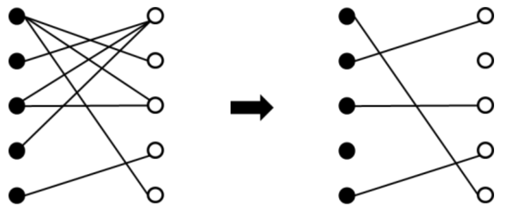


CHAPTER 9

EXERCISE 9.1. Construct a graph G = (N, A) containing a node for each period i along with a source node and a sink node. The capacity of the arc from the source node to the node representing period i is equal to the maximum amount that the trader can buy during period i. The lower bound of the arc from the node representing period i to the sink node is equal to the minimum amount which must be sold by the trader during period i. There also exists an arc (i, i+1) for all 1 ≤ i < T with a capacity equal to the maximum amount which can be held over from period i to period i+1. The flow on the arc (s, i) denotes the amount bought by the trader during period i, the flow on the arc (i, t) denotes the amount sold by the trader during period i, and the flow on the arc (i, i+1) denotes the amount held over from period i to period (i+1). The formulation is illustrated in Figure S9.1 for T = 4. An optimal circulation in the network would yield an optimal solution to the Entrepreneur's problem.

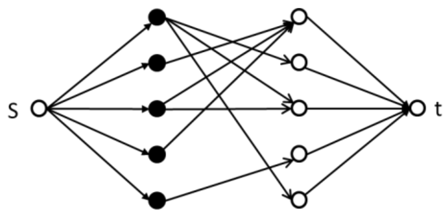
Figure S9.1

9. 给定n个黑点$\{b_1, b_2, \ldots, b_n\}$，以及$n$个白点$\{w_1, w_2, \ldots, w_n\}$，每个黑点可与若干个白点匹配，请使用最大流思想设计一个最大匹配算法，匹配最多的黑白点对，使得任意一个黑点最多与一个白点匹配。下图给出一最大匹配的例子。（共 10 分）



匹配前　　　匹配后

参考答案：增加虚拟节点s和t，构建类似如下有向图，使用最大流算法求解。



10. 给定两个字符串$A[1,2, \ldots, m]$和$B[1,2, \ldots, n]$，请设计方法求这两个字符串的最长公共子串。（共 10 分）

参考答案：

令 dp[i][j]表示 A 串中的以第 i - 1 个字符与 B 串中的以第 j - 1 个字符作为结尾的最长公共子串的长度，但是仅当 A[i-1] = B[j-1]且 dp[i][j] = dp[i -1][j -1] + 1;A[i-1] != B[j-1]时，以他们结尾的公共子串长度必然为 0；最后从 dp 中取出长度最大的值即为最长公共子串。

11. 有n个小朋友，每个小朋友手里拿着一张数字牌，第i个小朋友的数字牌为p_i，每张牌$p_i \leq B$。小朋友i和小朋友j能匹配当且仅当$p_i + p_j \leq B$。请设计贪心策略匹配最多对小朋友，分析是否可得最优解。

参考答案：（循环过程）

（1）将所有人的从小到大进行排序；
（2）从当前最小的人 i 开始考虑，找能跟其匹配的最大的人 j；
这样的方法是贪心的，因为它只是让“眼前的浪费”最少。
可以用反证法证明此策略的正确性：
当前最小的是 i，最大能与 i 匹配的是 j，
算法认为 i 和 j 匹配最好，若有人不认同该算法，那就让 i 和 k 匹配好了，显然 $pk<pj$，此时 $pj-pk$ 部分会浪费掉。