算法分析第6次作业

小组编号：23

本次作业负责人：李嘉琪

算法实现题6-1 答案：

6-1 最小长度电路板排列问题。

问题描述： 最小长度电路板排列问题是大规模电子系统设计中提出的实际问题。该问题的提法是，将 n 块电路板以最佳排列方案插入带有 n 个插槽的机箱中。n 块电路板的不同的排列方式对应不同的电路板插入方案。

设 $B=\{1, 2, \cdots, n\}$ 是 n 块电路板的集合。集合 $L=\{N_1, N_2, \cdots, N_m\}$ 是 n 块电路板的 m 个连接块。其中每个连接块 N_i 是 B 的一个子集，且 N_i 中的电路板用同一根导线连接在一起。在最小长度电路板排列问题中，连接块的长度是指该连接块中第 1 块电路板到最后 1 块电路板之间的距离。

试设计一个队列式分支限界法找出所给 n 个电路板的最佳排列，使得 m 个连接块中最大长度达到最小。

算法设计： 对于给定的电路板连接块，设计一个队列式分支限界法，找出所给 n 个电路板的最佳排列，使得 m 个连接块中最大长度达到最小。

数据输入： 由文件 input.txt 给出输入数据。第 1 行有 2 个正整数 n 和 m（$1 \leq m, n \leq 20$）。接下来的 n 行中，每行有 m 个数。第 k 行的第 j 个数为 0 表示电路板 k 不在连接块 j 中，为 1 表示电路板 k 在连接块 j 中。

结果输出： 将计算的电路板排列最小长度及其最佳排列输出到文件 output.txt。文件的第 1 行是最小长度；接下来的 1 行是最佳排列。

输入文件示例	输出文件示例
input.txt	output.txt
8 5	4
1 1 1 1 1	5 4 3 1 6 2 8 7
0 1 0 1 0	
0 1 1 1 0	
1 0 1 1 0	
1 0 1 0 0	
1 1 0 1 0	
0 0 0 0 1	
0 1 0 0 1	

算法描述：

解空间树：排列树，即B={1,2,......,n}中使得m个连接块中最大长度达到最小的最佳排列

解向量：$\{x_1, x_2, \ldots\ldots, x_n\}$，$x_i \in \{1, 2, \ldots\ldots, n\}$且任意$x_i \neq x_j$

变量说明：

`curLen`：记录当前m个连接块中的最大长度

`L[1:m]`：记录m个连接块中的当前长度

```
struct Count{
    int left;
    int right;
} ;// 连接块中对应的左右边界
```

`best[1:n]`：记录最优排列

`minLen`：记录m个连接块的最小长度

```
struct Node{
    int levels;// 当前解的位置
    Count count[m];// m个连接块的左右边界
    int result[n];// 当前解
}Node; 每个进入队列的结点
```

采用队列式分支限界法，遍历排列树，对当前某个扩展结点的子节点

遍历父节点`count[1:m]`更新子节点`count[1:m]`的值，子节点处在位置`level`

- 若子结点在Li中且Li无结点$|Li|=0$，则更新$count[i].left=count[i].right=level$
- 若子结点在Li中且结点$|Li|\geq 1$，则更新$count[i].right=level$
- 若子结点不在$L[i]$中，则不更新$count[i]$，与父节点一致

  在遍历L中计算$count[i].right-count[i].left$，求出其中最大长度$curLen$作为剪枝

  若$curLen \geq minLen$，则子节点不加入队列

  若$curLen< minLen$，则当前子节点入队列，更新$minLen$，取出队列的下一个结点

当遍历到叶子结点时，得到一种更好的排列，更新最小长度$minLen$并更新最优解$best$，算法继续

直到队列为空，算法结束

算法分析：

考虑剪枝效果最差的情况，排列树的结点数量为

$$n+n(n-1)+n(n-1)(n-2)+......+n(n-1)(n-2)......1$$

计算剪枝的时间复杂性为$O(m)$

时间复杂度为

$$O(m)\times[n+n(n-1)+n(n-1)(n-2)+......+n(n-1)(n-2)......1]=O(mn!)$$

本题分工：小组共同讨论，黄勖编写

算法实现题6-2 答案：

6-2 最小权顶点覆盖问题。

问题描述：给定一个赋权无向图 $G=(V, E)$，每个顶点 $v \in V$ 都有权值 $w(v)$。如果 $U \subseteq V$，且对任意 $(u, v) \in E$ 有 $u \in U$ 或 $v \in U$，就称 U 为图 G 的一个顶点覆盖。G 的最小权顶点覆盖是指 G 中所含顶点权之和最小的顶点覆盖。

算法设计：对于给定的无向图 G，设计一个优先队列式分支限界法，计算 G 的最小权顶点覆盖。

数据输入：由文件 input.txt 给出输入数据。第 1 行有 2 个正整数 n 和 m，表示给定的图 G 有 n 个顶点和 m 条边，顶点编号为 1, 2, …, n。第 2 行有 n 个正整数表示 n 个顶点的权。接下来的 m 行中，每行有 2 个正整数 u 和 v，表示图 G 的一条边 (u, v)。

结果输出：将计算的最小权顶点覆盖的顶点权之和以及最优解输出到文件 output.txt。文件的第 1 行是最小权顶点覆盖顶点权之和；第 2 行是最优解 x_i（$1 \leq i \leq n$），$x_i=0$ 表示顶点 i 不在最小权顶点覆盖中，$x_i=1$ 表示顶点 i 在最小权顶点覆盖中。

输入文件示例	输出文件示例
input.txt	output.txt
7 7	13
1 100 1 1 1 100 10	1 0 1 1 0 0 1
1 6	
2 4	
2 5	
3 6	
4 5	
4 6	
6 7	

算法描述：

解空间树：子集树

解向量：$\{x_1, x_2, ..., x_n\}$，其中$x_i \in \{0,1\}$ 且 $i \in \{1,2,......,n\}$

优先级：结点的优先级 = 当前已经选择结点的权和 + 选择当前结点的权

即子节点的优先级 = 父节点的优先级 + $x_i \times w_i$

剪枝策略：使用贪心策略循环每次选择当前最小的权的点并去掉相关的所有边直到所有的边均去掉后得到上界限$bound$

采用小根堆方式组织优先队列遍历子集树，每次取出队列首元素作为扩展结点，判断当前扩展结点表示的解是否构成顶点覆盖

- 若构成顶点覆盖，则找到最优值和最优解，算法结束
- 若不构成顶点覆盖，则从当前扩展结点扩展子节点，计算子结点的优先级

  若子节点的优先级 $> bound$，则子节点不能加入队列

  若子节点的优先级 $\leq bound$，则将子节点加入队列

  算法继续

算法分析：

结点数量为$2^{n+1}-2$

计算每个结点的优先级时间复杂度$O(1)$

贪心策略求上界的时间复杂度为$O(n^2)$

时间复杂度为

$$(2^{n+1}-2)\times O(1)\times O(n^2)=O(n^2 2^{n+1})$$

本题分工：小组共同讨论，李嘉琪编写

算法实现题6-4 答案：

6-4　最小重量机器设计问题。

问题描述： 设某一机器由 n 个部件组成，每种部件都可以从 m 个不同的供应商处购得。设 w_{ij} 是从供应商 j 处购得的部件 i 的重量，c_{ij} 是相应的价格。设计一个优先队列式分支限界法，给出总价格不超过 d 的最小重量机器设计。

算法设计： 对于给定的机器部件重量和机器部件价格，设计一个优先队列式分支限界法，计算总价格不超过 d 的最小重量机器设计。

数据输入： 由文件 input.txt 给出输入数据。第 1 行有 3 个正整数 n、m 和 d。接下来的 $2n$ 行，每行 n 个数。前 n 行是 c，后 n 行是 w。

结果输出： 将计算的最小重量，以及每个部件的供应商输出到文件 output.txt。

输入文件示例	输出文件示例
input.txt	output.txt
3 3 4	4
1 2 3	1 3 1
3 2 1	
2 2 2	
1 2 3	
3 2 1	
2 2 2	

算法描述：

解空间树：m叉树，即求n个位置，每个位置的取值为$1,2,......,m$的排列

优先级：子结点优先级=父节点优先级+子结点表示的重量+剩余未选择位置中重量最小值之和

采用小根堆组织优先队列，结点优先级最小的优先取出处理

采用优先队列方式遍历m叉树时，每次取出队头元素作为扩展结点，对每个扩展结点的子节点判断其是否可以加入优先队列：

- 若扩展结点为叶子结点，即当前结点处在n层，则得到最小重量和最优解
- 若扩展结点不为叶子节点，即当前结点处在1~n-1层，则：

  若将扩展结点的子节点添加到当前解后总价格超过d，则子节点不能加入优先队列；

  若当前扩展结点的子节点添加到当前解后总价格不超过d，则计算当前子节点的优先级并将其加入队列

算法分析：

考虑最坏情况下，每个结点计算优先级(求剩余未选择重量最小值之和)的时间复杂度为$O(mn)$

$n-1$层m叉树的结点数量为$\frac{1-m^n}{1-m}$

$$\frac{1-m^n}{1-m}O(mn)=O(m^n n)$$

本题分工：小组共同讨论，黄勖编写

算法实现题6-5 答案：

6-5 运动员最佳配对问题。

问题描述：羽毛球队有男女运动员各 n 人。给定 2 个 $n \times n$ 矩阵 $\boldsymbol{P}$ 和 $\boldsymbol{Q}$。$P[i][j]$是男运动员 i 和女运动员 j 配对组成混合双打的男运动员竞赛优势；$Q[i][j]$是女运动员 i 和男运动员 j 配合的女运动员竞赛优势。由于技术配合和心理状态等因素影响，$P[i][j]$不一定等于 $Q[j][i]$。男运动员 i 和女运动员 j 配对组成混合双打的男女双方竞赛优势为 $P[i][j] \times Q[j][i]$。设计一个算法，计算男女运动员最佳配对法，使各组男女双方竞赛优势的总和达到最大。

算法设计：设计一个优先队列式分支限界法，对于给定的男女运动员竞赛优势，计算男女运动员最佳配对法，使各组男女双方竞赛优势的总和达到最大。

数据输入：由文件 input.txt 给出输入数据。第 1 行有 1 个正整数 n（$1 \leqslant n \leqslant 20$）。接下来的 $2n$ 行，每行 n 个数。前 n 行是 p，后 n 行是 q。

结果输出：将计算的男女双方竞赛优势的总和的最大值输出到文件 output.txt。

输入文件示例	输出文件示例
input.txt	output.txt
3	52
10 2 3	
2 3 4	
3 4 5	
2 2 2	
3 5 3	
4 5 1	

算法描述：

解空间树：n排列树

优先级：当前已经完成的配对的竞赛优势和 + 剩余未配对中竞赛优势最大的和

子节点的优先级 = 父节点的优先级别 + 当前配对竞赛优势 + 剩余未配对最大竞赛优势之和

剪枝策略：使用贪心算法对n个男生中男生i，计算当前最大的$P[i][j]\times Q[i][j]$（j未被选择）之和作为下界$bound$

采用大根堆组织优先队列，每次优先处理优先级数值大的结点

遍历n排列树，每次取出优先级别最大的结点作为扩展结点，对每个扩展结点的子节点：

- 若子节点为叶子节点，即到达第n层，得到竞赛优势总和的最优值，算法结束
- 若子节点不为叶子结点，即在1~n-1层次，计算当前结点的优先级
  若结点的优先级$< bound$，则当前结点应舍弃，不加入优先队列
  若结点的优先级$\geq bound$，则当前结点加入优先队列
  算法继续

算法分析:

计算结点的优先级时间复杂度为$O(n^2)$

结点数量为

$$n + n(n-1) + n(n-1)(n-2) + \ldots\ldots + n(n-1)(n-2)\ldots\ldots 1$$

因此时间复杂度为

$$O(nn!)$$

本题分工：小组共同讨论，李嘉琪编写

算法实现题6-10 答案:

6-10　世界名画陈列馆问题。

问题描述： 世界名画陈列馆由 $m \times n$ 个排列成矩形阵列的陈列室组成。为了防止名画被盗，需要在陈列室中设置警卫机器人哨位。除了监视所在的陈列室，每个警卫机器人还可以监视与它所在的陈列室相邻的上、下、左、右 4 个陈列室。试设计一个安排警卫机器人哨位的算法，使名画陈列馆中每个陈列室都在警卫机器人的监视下，且所用的警卫机器人数最少。

算法设计： 设计一个优先队列式分支限界法，计算警卫机器人的最佳哨位安排，使名画陈列馆中每个陈列室都在警卫机器人的监视下，且所用的警卫机器人数最少。

数据输入： 由文件 input.txt 给出输入数据。第 1 行有 2 个正整数 m 和 n（$1 \le m, n \le 20$）。

结果输出： 将计算的警卫机器人数及其最佳哨位安排输出到文件 output.txt。文件的第 1 行是警卫机器人数；接下来的 m 行中每行 n 个数，0 表示无哨位，1 表示有哨位。

输入文件示例	输出文件示例
input.txt	output.txt
4 4	4
	0 0 1 0
	1 0 0 0
	0 0 0 1
	0 1 0 0

算法描述:

解空间树：子集树

解向量：$\{x_1, x_2, \ldots\ldots, x_{m\times n}\}$ 且 $x_i \in \{0,1\}, i \in \{1, 2 \ldots\ldots, m \times n\}$

优先级：某个结点的优先级 = 当前解的路径中1的个数

当前结点的优先级别 = 父结点的优先级 + 当前结点表示的值

剪枝策略：考虑上界剪枝

任何3x3的方格需要3个一定可以构成监视

任何3x2或者2x2的方格需要3个一定可以构成监视

任何nx1的方格需要n/3 + n % 3个一定可以构成监视

根据输入的mxn的值将其进行划分：

- 优先划分为多个3x3的块
- 根据剩余的块拼凑成3x2或者2x2的块
- 最后确定nx1块的数量

计算上界up

采用优先队列式分支限界法，采用小根堆，从根节点开始遍历子集树，每次取出当前优先级值最小的作为扩展结点：

- 若扩展结点构成所有陈列室都被监视，则得到最少的机器人人数和最优解，算法结束
- 若扩展结点不能构成所有陈列室被监视，则扩展子节点，并计算子节点优先级

  若子节点优先级 > up，则舍弃子节点，不加入队列

  若子节点优先级 <= up，则子节点加入队列

  算法继续

算法分析：

考虑最坏情况下结点数量为

$$2^1+2^2+......+2^{m\times n}=2(2^{m\times n}-1)$$

计算每个结点的代价为O(1)

因此时间复杂度约为

$$O(2^{m\times n})$$

本题分工：小组共同讨论，黄勖编写