第七八章 回溯与分支限界法作业

作业1

- 回溯法和分支限界法的状态空间生成方式有何不同？简述两种方法的基本思想。
- 状态空间生成方式
  - 回溯法: 当前的E-结点R 一旦生成一个新的儿子结点C, 这个C结点就变成一个新的E-结点, 当检测完了子树C后, R结点就再次成为E-结点, 生成下一个儿子结点。
  - 分支-限界方法：当前结点一旦成为E-结点，就一直处理到变成死结点为止。其生成的儿子结点加入到活结点表中，然后再从活结点表中选择下一个新的E-结点。

- 回溯法：
  - 回溯法从根结点出发，按深度优先策略搜索解空间树。判断E-结点是否包含问题的解。如果不包含，则跳过以该结点为根的子树，逐层向其祖先结点回溯。否则就进入该子树，继续按深度优先的策略进行搜索。
- 分支限界法：
  - 分支限界法从根结点出发，生成当前E-结点全部儿子之后，再选择新的活结点成为E-结点，重复上述过程。为提高算法效率，会使用限界函数和成本估计函数等进行剪枝。

作业2

- 回溯法的效率估计问题

(1) 考虑每一行每一列各不相同，求6皇后问题静态的状态空间树的结点总数；
(2) 按图1所示，采用蒙特卡罗方法估计不受限结点数（即实际扩展的结点数）；
(3) 根据(2)的结果和(1)比较，计算回溯算法的不受限结点数占结点总数的比例。

	Q				
				Q	
		Q			
					Q
			Q		

图 1. 6 皇后问题的一个状态

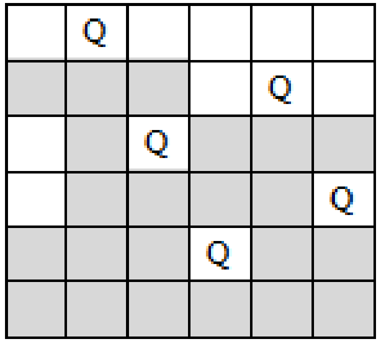
- 第(1)问：

$$1+6+(6\times5)+(6\times5\times4)+(6\times5\times4\times3)+(6\times5\times4\times3\times2)+(6\times5\times4\times3\times2\times1)$$

$$=1957$$

- 第(2)问：

分析得到每行不受限结点个数(6,3,2,2,1)



$$1+6+(6\times3)+(6\times3\times2)+(6\times3\times2\times2)+(6\times3\times2\times2\times1)=205$$

- 第(3)问：

205/1957=10.5%

作业3

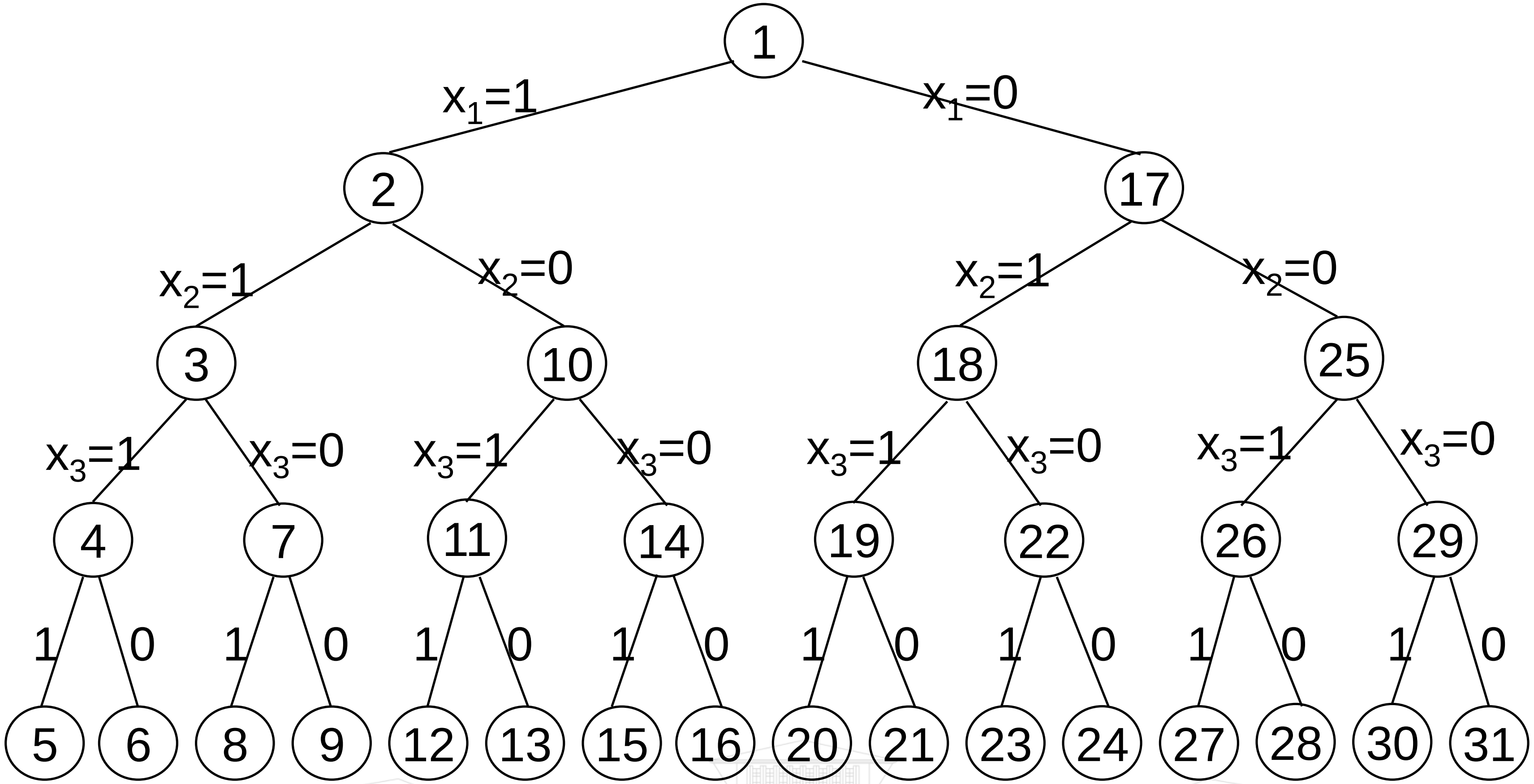
- 子集和问题：

(1) 画出定长元组（n-元组）表示下，n=4时子集和问题的状态空间树，状态结点按深度优先顺序编号。

(2) 给出n-元组表示下，子集和问题的限界函数（剪枝的条件）。

(3) 对于W=(2,5,3,10) 和M=15的问题实例，在1)的基础上，利用2)的限界函数，画出回溯法剪枝后的状态空间树，状态结点序号同1）。

(1) 画出定长元组（n-元组）表示下，n=4时子集和问题的状态空间树，状态结点按深度优先顺序编号。



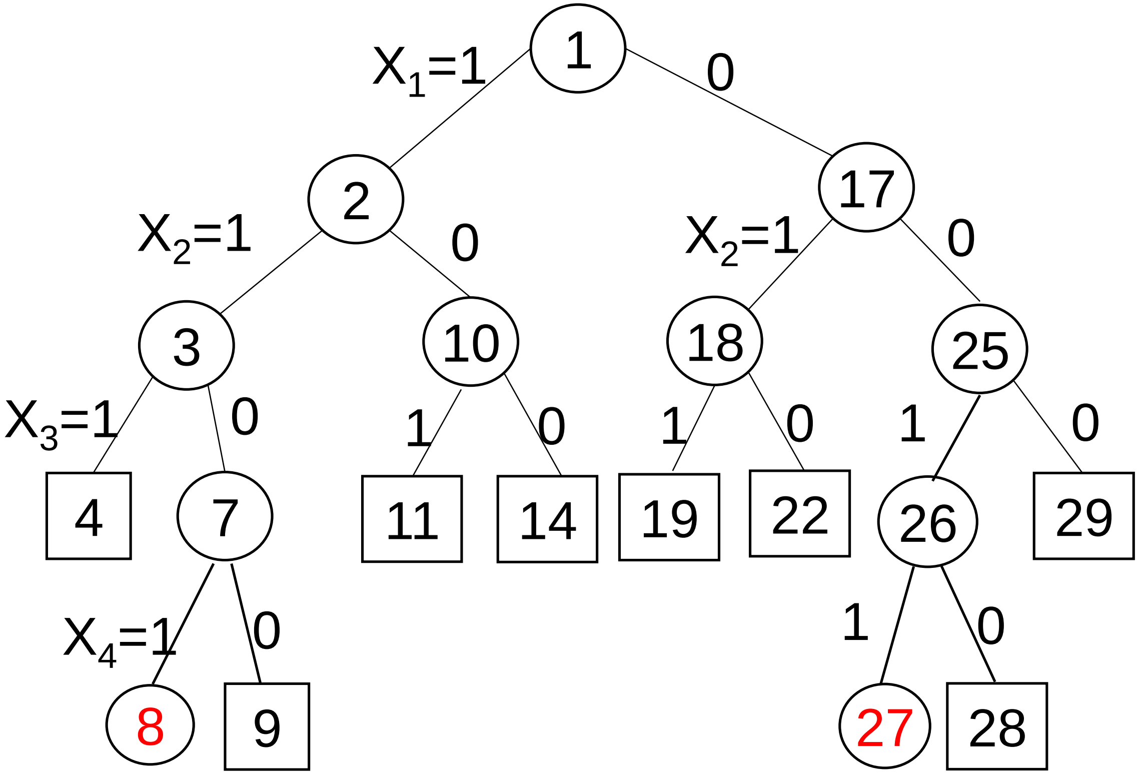
(2) 给出n-元组表示下，子集和问题的限界函数（剪枝的条件）。
答：将W(i)按非降次序排列

- 限界函数B_k(X(1),…,X(k))=true，当且仅当：

$$\underbrace{\sum_{i=1}^{k} W(i)X(i)}_{s} + \underbrace{\sum_{i=k+1}^{n} W(i)}_{r} \geq M \quad 且 \quad \sum_{i=1}^{k} W(i)X(i) + W(k+1) \leq M$$

3）n-元组表达下，
对于W=(2,5,3,10) 和M=15的问题实例，
按照数值从小到达排序为W=(2,3,5,10)

结点编号	s	r	W(i+1)	B值
2	2	18	3	T
3	5	15	5	T
4	10	10	10	F
7	5	10	10	T
8(答案)	15	0	0	T
9	5	0	0	F
10	2	15	5	T
11	7	10	10	F
14	2	10	10	F
17	0	18	3	T
18	3	15	5	T
19	8	10	10	F
22	3	10	10	F
25	0	15	5	T
26	5	10	10	T
27(答案)	15	0	0	T
28	5	0	0	F
29	0	10	10	F



作业4

- 下图是15谜问题

(1) 判定图中初始状态是否能达到目标状态？（需要计算过程）

(2) 设$\hat{c}(X)=f(X)+\hat{g}(X)$。$f(X)$=根到结点X的路径长度，$\hat{g}(X)$=不在其目标位置的非空白牌数目，用LC检索法，画出从初始状态到达目标状态的状态空间树，并标出树中每个状态节点的$\hat{c}$值。

1	2		4
5	6	3	8
9	10	7	11
13	14	15	12

初始状态

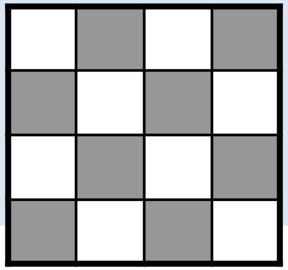
1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	

目标状态

定理：当且仅当 Σ LESS (i)+X是偶数时，目标状态可由此初始状态到达。

LESS (i)是使牌j<牌i，且POSITION(j)> POSITION(i)的j的数目.

如果空格在图的阴影位置中的某一格处，则令X=1；否则X=0。



1	2		4
5	6	3	8
9	10	7	11
13	14	15	12

初始状态

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	

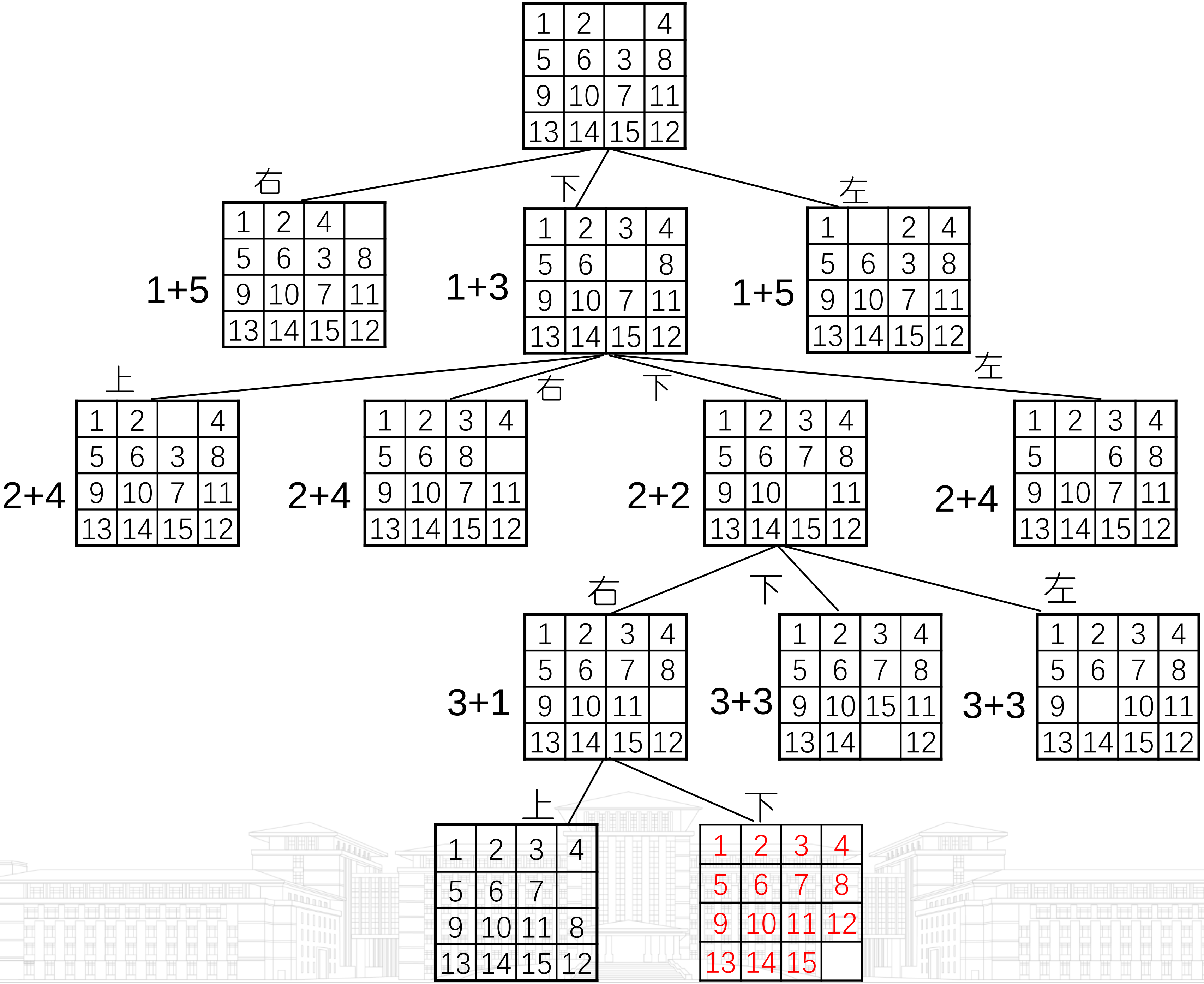
目标状态

答：1) 求LESS(i)，得到∑Less(i)=22

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
LESS	0	0	0	1	1	1	0	1	1	1	0	0	1	1	1	13

2) 对照空格位，X=0，则∑Less(i)+0=22，为偶数。

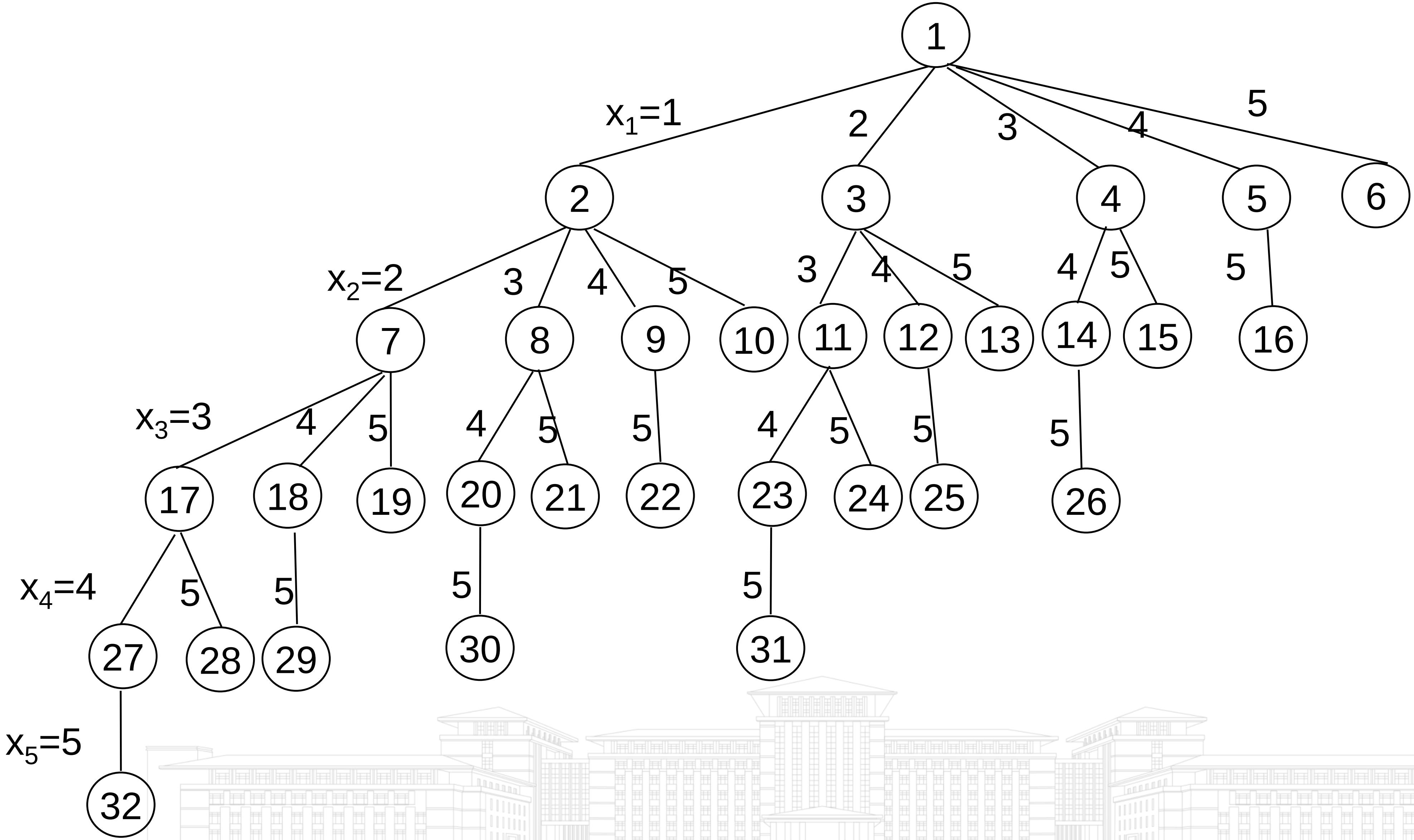
因此，目标状态可达。


右
下
左
1+5
1+3
1+5
上
右
下
左
2+4
2+4
2+2
2+4
右
下
左
3+1
3+3
3+3
上
下

作业5

- 给定带期限的作业调度问题：
  - n=5，(p1,d1,t1)=(6,3,2), (p2,d2,t2)=(3,1,1), (p3,d3,t3)=(4,3,2), (p4,d4,t4)=(8,2,1), (p5,d5,t5)=(5,3,1)。假定问题采用不定长元组表示。
  - (1) 请给出n=5的静态树，状态结点采用层次遍历编号。
  - (2) 请给出ĉ(X)和u(X)的定义。
  - (3) 分别画出FIFOBB算法和LCBB算法生成的动态树，并标出对应最优解的答案结点和给出罚款数。

(1) 请给出n=5的静态树，状态结点采用层次遍历编号。



(2) 请给出$\hat{c}(X)$和$u(X)$的定义。

- 设Sx是根结点到达结点X时选中的作业集合，$m=\max\{i|i\in S_X\}$,
  - $\hat{c}(X)=\sum\limits_{\substack{i<m \\ i\notin S_X}} p_i$
  - $u(X)=\sum\limits_{i\notin S_X} p_i$
- U 整棵树中C(T) 上界(当前找到的最小罚款值)
  - 初值$U=\infty$或$U=\sum\limits_{1\le i\le n} p_i$

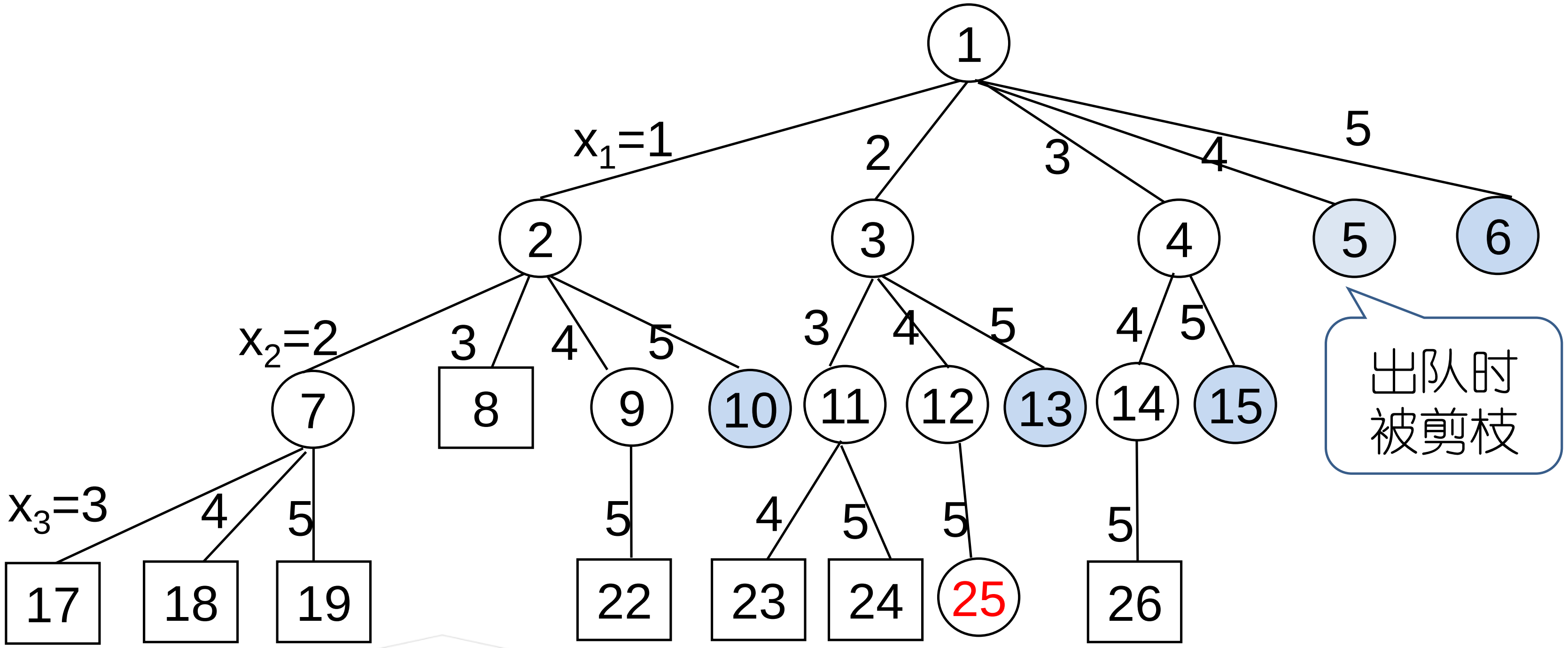
结点编号	$\hat{c}(X)$	$u(X)$	
2	0	20	U=20
3	6	23	
4	9	22	
5	13	18	U=18
6	21	21	
7	0	17	U=17
8			
9	7	12	U=12
10	15	15	
11	6	19	
12	10	15	
13	18	18	
14	9	14	
15	17	17	

$(p_1,d_1,t_1)=(6,3,2)$, $(p_2,d_2,t_2)=(3,1,1)$, $(p_3,d_3,t_3)=(4,3,2)$, $(p_4,d_4,t_4)=(8,2,1)$, $(p_5,d_5,t_5)=(5,3,1)$

(3) FIFOBB算法生成的动态树。

蓝色表示$\hat{c}(X) \geq U$ 舍弃

灰色表示B函数剪枝，对应方形结点



结点编号	$\hat{c}(X)$	$u(X)$
17		
18		
19		
22		
23		
24		
25	10	10
26		

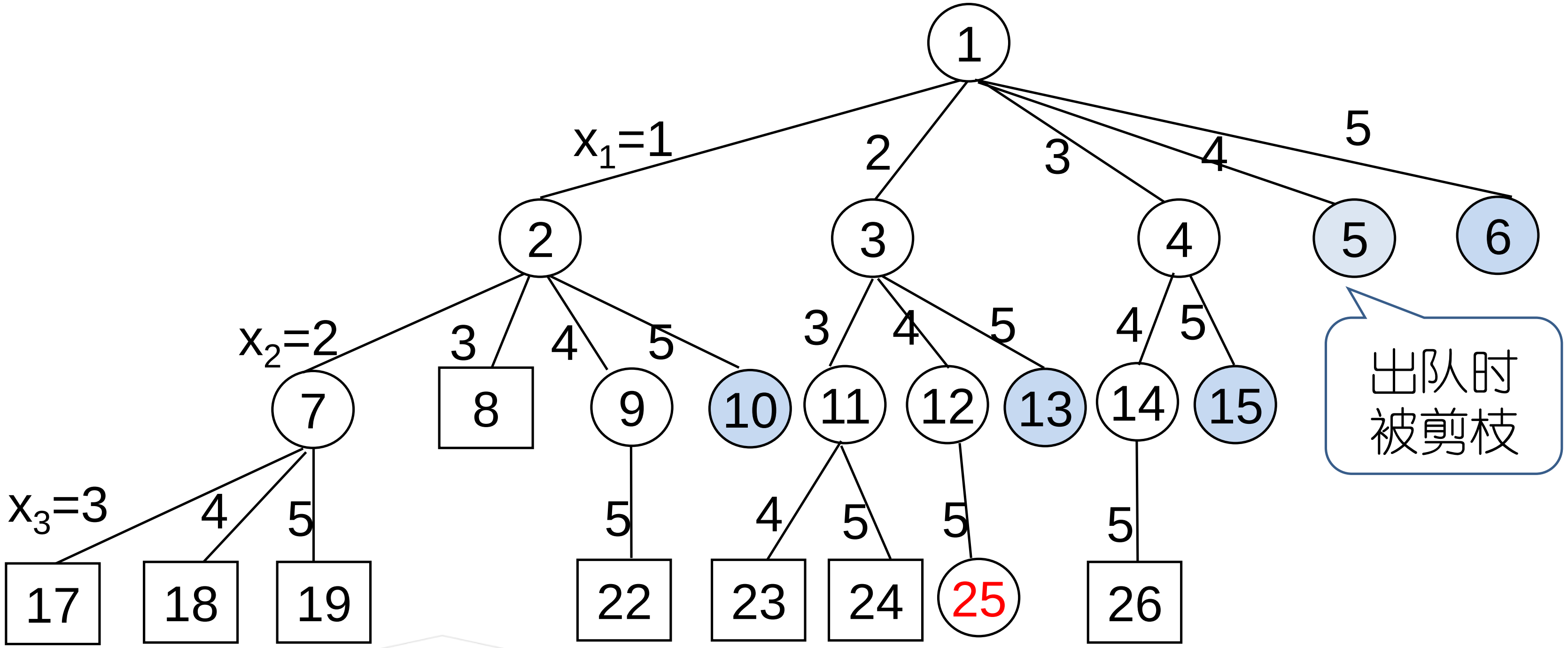
U=10

$(p_1,d_1,t_1)=(6,3,2)$, $(p_2,d_2,t_2)=(3,1,1)$, $(p_3,d_3,t_3)=(4,3,2)$, $(p_4,d_4,t_4)=(8,2,1)$, $(p_5,d_5,t_5)=(5,3,1)$

(3) FIFOBB算法生成的动态树。

蓝色表示$\hat{c}(X) \geq U$ 舍弃

灰色表示B函数剪枝，对应方形结点



一共23个结点，最优解的答案结点25，最小罚款数10。

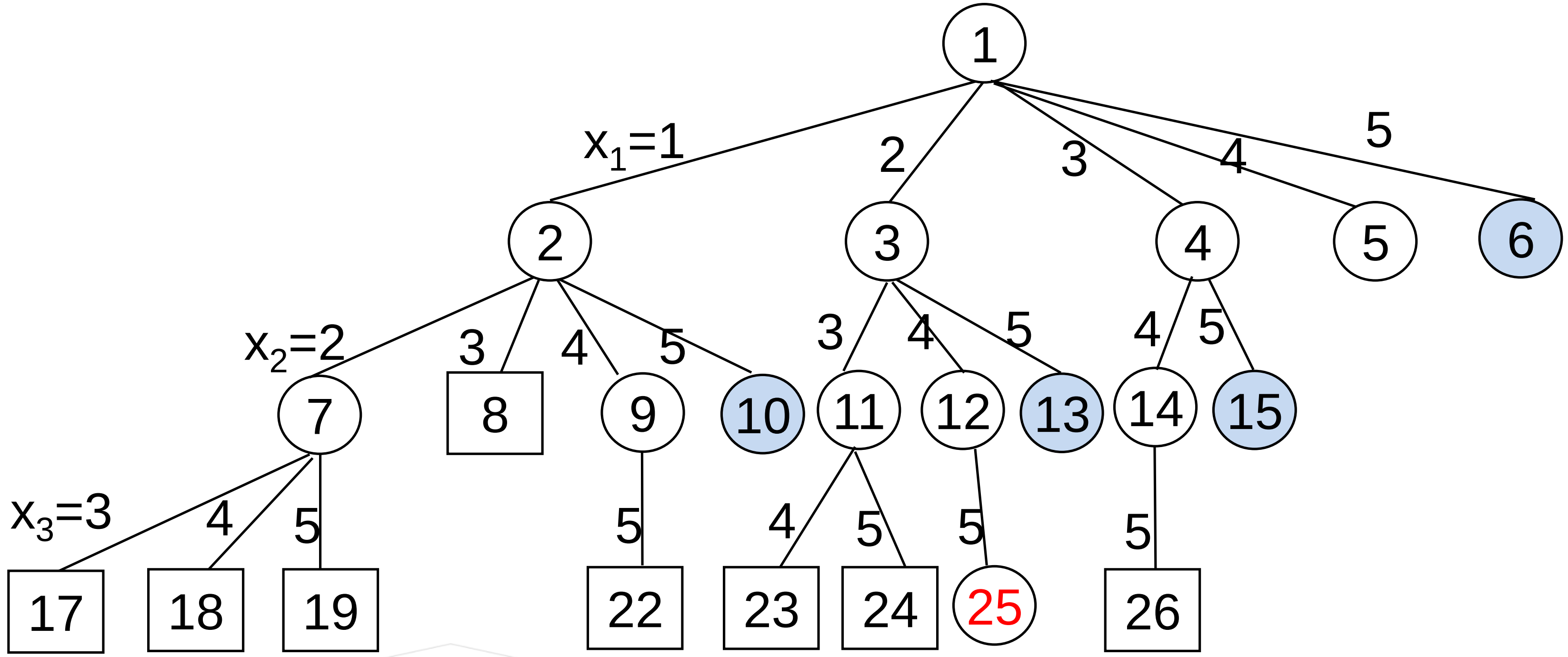
E结点	结点编号	ĉ(X)	u(X)	
1	2	0	20	U=20
	3	6	23	
	4	9	22	
	5	13	18	U=18
	6	21	21	
2	7	0	17	U=17
	8			
	9	7	12	U=12
	10	15	15	
7	17			
	18			
	19			
3	11	6	19	
	12	10	15	

(p1,d1,t1)=(6,3,2), (p2,d2,t2)=(3,1,1), (p3,d3,t3)=(4,3,2), (p4,d4,t4)=(8,2,1), (p5,d5,t5)=(5,3,1)

(3) LCBB算法生成的动态树。

蓝色表示ĉ(X) ≥ U 舍弃

灰色表示B函数剪枝，对应方形结点



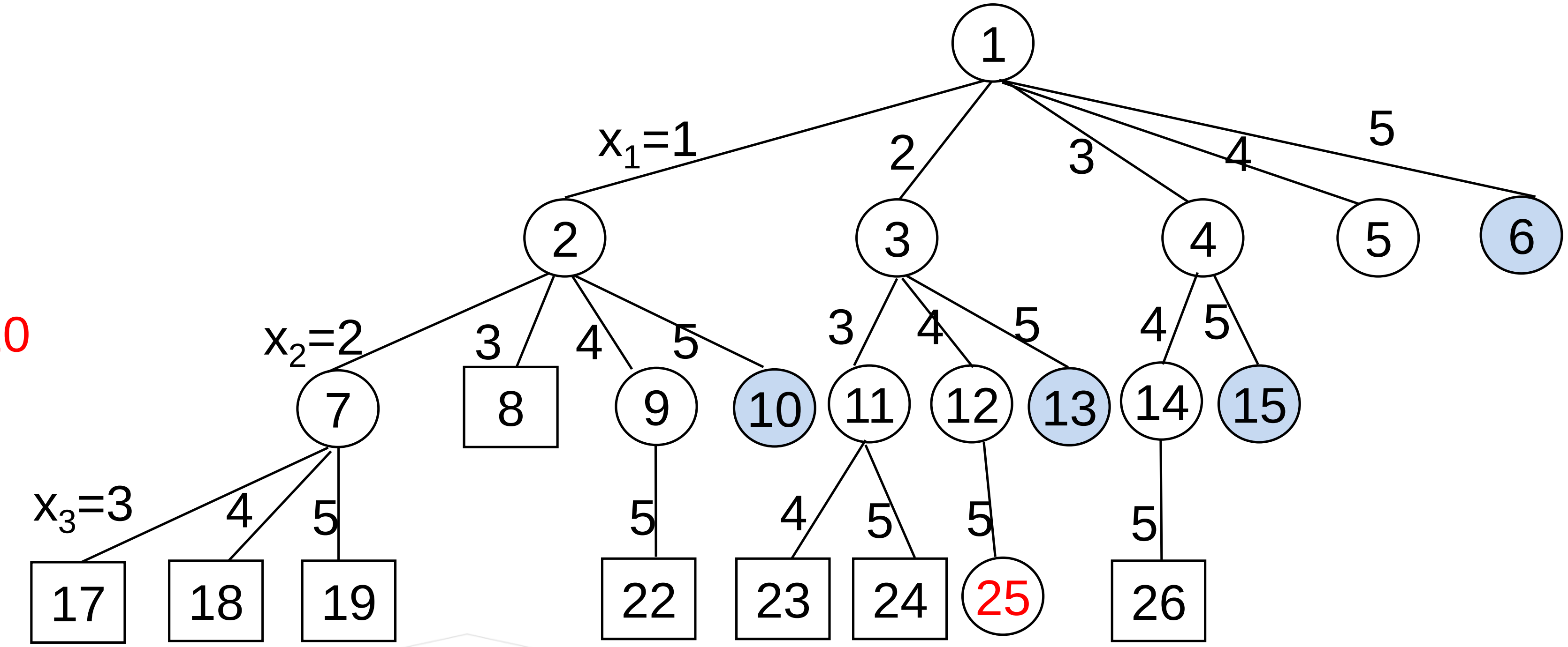
E结点	结点编号	$\hat{c}(X)$	$u(X)$	
	13	18	18	
11	23			
	24			
9	22			
4	14	9	14	
	15	17	17	
14	26			
12	25	10	10	U=10

$(p_1,d_1,t_1)=(6,3,2)$, $(p_2,d_2,t_2)=(3,1,1)$, $(p_3,d_3,t_3)=(4,3,2)$, $(p_4,d_4,t_4)=(8,2,1)$, $(p_5,d_5,t_5)=(5,3,1)$

(3) LCBB算法生成的动态树。

蓝色表示$\hat{c}(X) \geq U$ 舍弃

灰色表示B函数剪枝，对应方形结点



一共23个结点，最优解的答案结点25，最小罚款数10。

作业6

- 分配问题：设有n件工作分配给n个人，工作i分配给第j个人所需的费用为c(i,j)。已知二维数组c(n,n)，现在请设计一个最佳工作分配方案，使总费用达到最小。

  （1）确定问题的元组表达形式、显式约束条件和隐式约束条件

  （2）给出n=4的解空间树

  （3）设计算法，给出问题的最优解和最佳工作分配方案。

分析：已知n件工作分配给n个人，每个人都会分配到一件工作。
(1) 表达式分析：将c(n,n)想像成棋盘，则问题类似于于n-皇后问题。
(2) 算法分析：求最优解问题适合采用分支限界法求解。

（1）确定问题的元组表达形式、显式约束条件和隐式约束条件

- n元组(x_1, … , x_n)，表示作业i分配给第x_i个人
- 显式约束条件：$1 \le x_i \le n$
- 隐式约束条件：x_i各不相同
- 目标函数：$\min\sum c(i, x_i)$，i=1,..,n

（2）给出n=4的解空间树

- 不同行不同列时，解空间树叶结点个数4！，叶结点是问题的解结点，结点编号按层次顺序编号，答案略。

基于分支限界法考虑问题

(3）设计算法，给出问题的最优解和最佳工作分配方案。

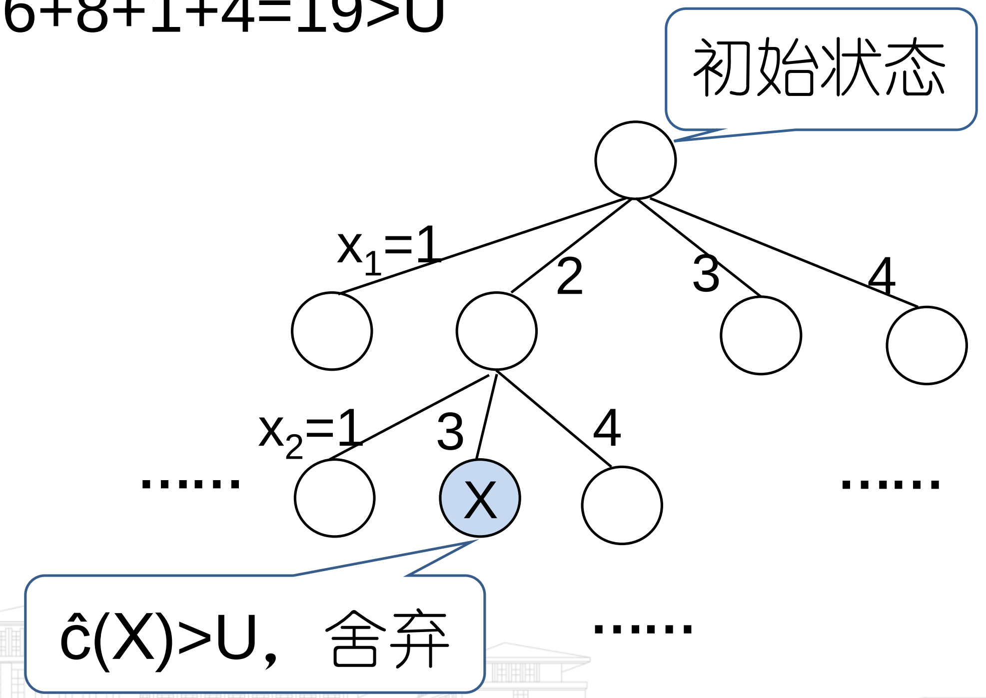
设计思想参考，设$x_1, \ldots, x_k$是根到X的路径：

- 1）成本上界U
  - 初值：从作业1到作业n，贪心法获得的可行解
  - 修改：当前若X是可行解，且成本值小于U，则赋值给U。
- 2）成本下界$\hat{c}(X)=f(X)+\hat{g}(X)$
  - $f(X)=\sum_{1\le i\le k} c(i,x_i)$
  - $\hat{g}(X)=\sum_{\substack{k<i\le n \\ 1\le j\le n}} \min\{c(i,j)\}$，即工作k+1到工作n中最小成本之和。
- 3）基于FIFO-分支限界法或LC-分支限界法实现
  - 当$\hat{c}(X)>U$时，X舍弃

问题举例：n=4，作业成本如下图表格所示。

- U初值：5+2+3+4=14，见表格中红色数字。
- 每行最小值：c(1,3),c(2,1),c(3,3),c(4,4)
- $\hat{c}(X) = \underbrace{c(1,2)+c(2,3)}_{\text{根到X的成本}}+\underbrace{c(3,3)+c(4,4)}_{\text{未来可能的最小成本}}=6+8+1+4=19>U$

人员	1	2	3	4
工作1	9	6	5	7
2	2	4	8	6
3	7	3	1	9
4	8	7	8	4



作业7

- 最大团问题：考虑无向图 G=<V, E>，已知G的完全子图G′=<V′, E′>, 有 V′ ⊆V, E′ ⊆E，G′中任意两个顶点之间都存在一条边相连。完全子图G′也称为G的团，团的大小用所含的结点数表示。请给出G的一个最大团。提示：可转化为判定问题。

（1）确定判定问题的元组表达形式、显式约束条件和隐式约束条件。

（2）设G的顶点数为5，判定是否存在大小为3的团，请给出该规模的解空间树。

（3）设计算法，给出问题的最大团和它的顶点集合。

方法1分析：问题转化为判定问题，即G中是否存在大小为m的团。
(1) 表达式分析：类似于子集和问题或0/1背包问题，判断是否能选出m个顶点。
(2) 算法分析：回溯法找到一个可行解。

(1) 确定判定问题的元组表达形式、显式约束条件和隐式约束条件。

- m元组$(x_1, \dots, x_m)$，m表示团的大小
- 显式约束条件：$x_i \in \{1,\dots,n\}$，1..n对应G中的顶点
- 隐式约束条件：任意两个x_i都不相同，且x_i之间都存在边相连， $x_i \le x_{i+1}$，$1 \le i < m$

(2) 设G的顶点数为5，判定是否存在大小为3的团，请给出该规模的解空间树。在静态树构造时考虑以下三个条件，使解空间个数为C^m_n个，当前规模为10个

- 1) m元组$(x_1, \dots, x_m)$, $x_i \in \{1,\dots,n\}$，1..n对应G中的顶点，i=1,..,m
- 2) 任意两个x_i都不相同
- 3) $x_i \le x_{i+1}$, $1 \le i < m$

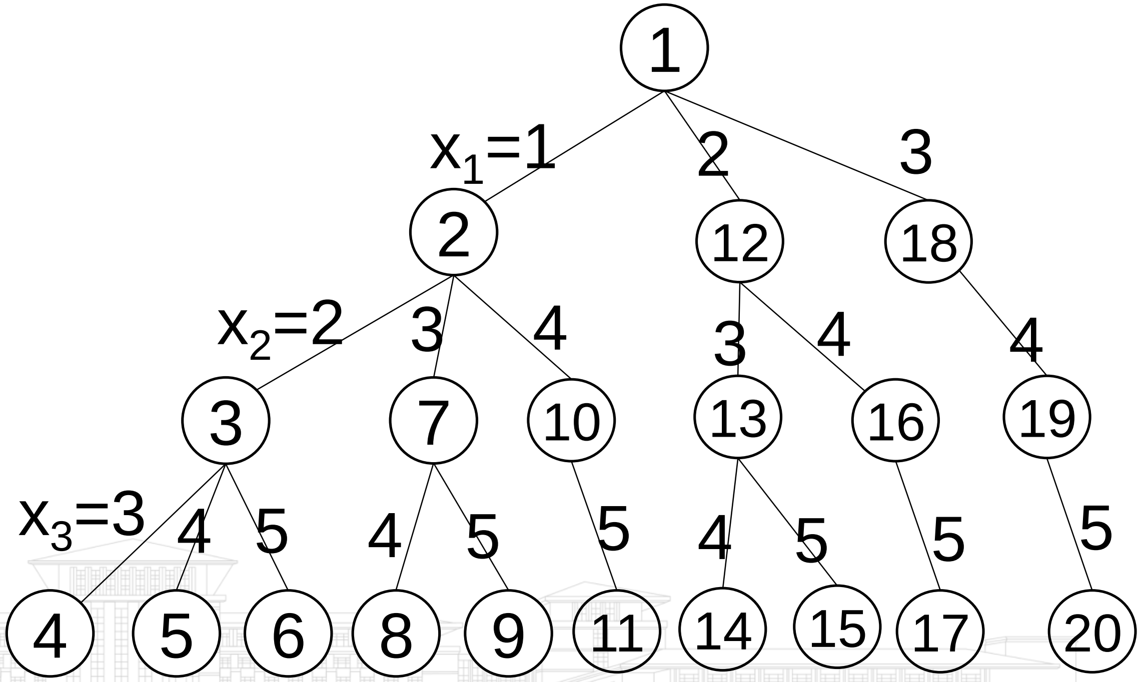
构造办法：对于$x_i=k$，则$x_{i+1}=k+1,\dots,n-(m-i)+1$，i=0,..,m-1

约束函数B_i：顶点x_i与$x_1, \dots, x_{i-1}$能成团，即有边相连

i=0，子结点取值到n-(m-i)+1=3

i=1，子结点取值到n-(m-i)+1=4

i=2，子结点取值到n-(m-i)+1=5



（3）设计算法，给出问题的最大团和它的顶点集合。

- 基于回溯算法设计，详情略。
- 对于n个顶点的图G，分析最坏情况下时间复杂度：
  - 1）解空间树叶结点个数：C^m_n个m-元组
  - 2）B判断叶结点是否是可行解，即x_m与x_1, … , x_{m-1}是否成团：m-1
  - 3）求最大团最多计算

$$\sum_{m=1}^{n} C^m_n(m-1) = O(n2^n)$$

指数级

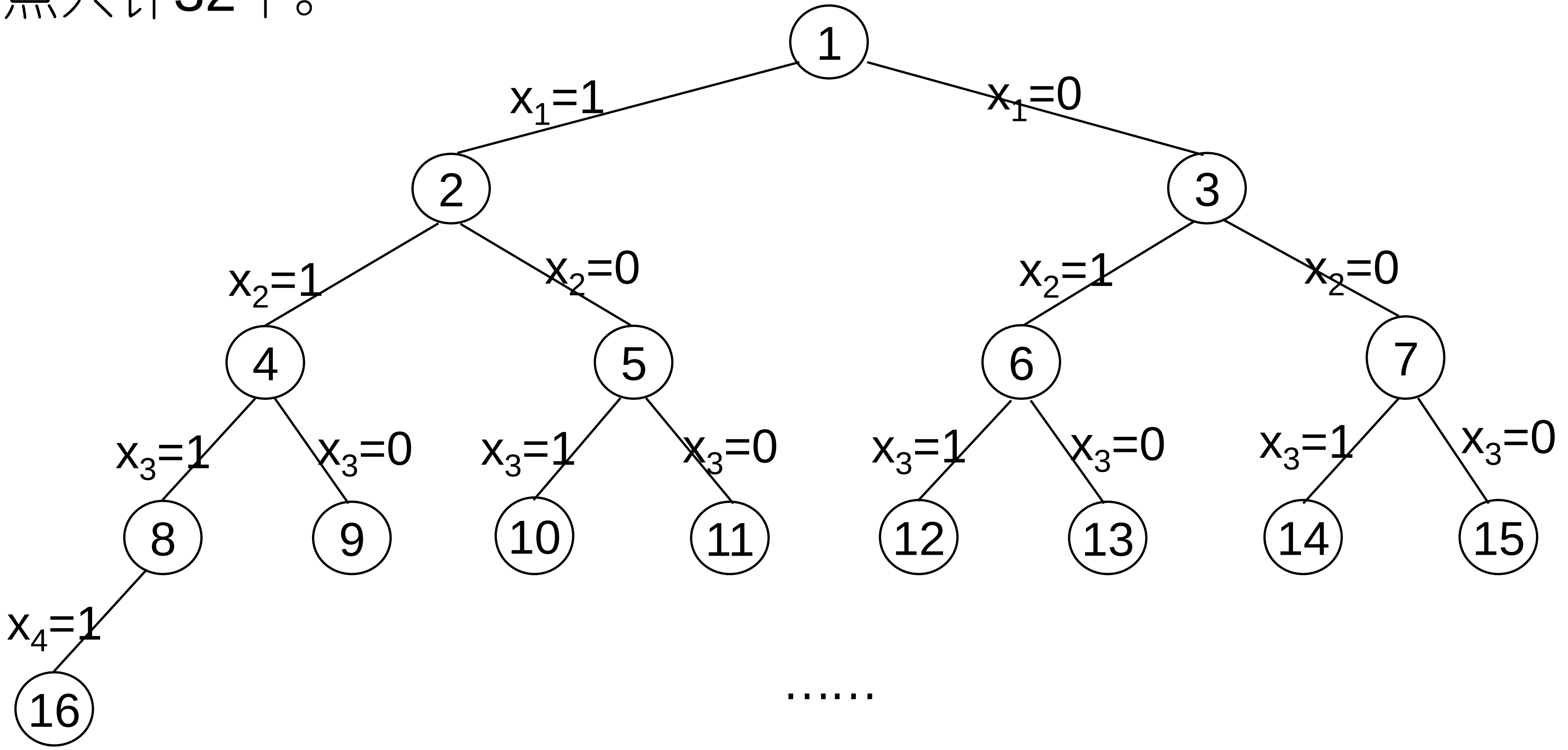
方法2分析：也可以直接求最大团问题。
(1) 表达式分析：类似于0/1背包问题。
(2) 算法分析：利用分支限界法求最优解。

(1) 确定元组表达形式、显式约束条件和隐式约束条件。

- n元组(x_1, … , x_n)，n表示G中顶点个数
- 显式约束条件：$x_i \in \{0,1\}$，1表示团选中顶点i，否则舍弃。
- 隐式约束条件：任意两个x_i=1对应的顶点i之间都存在边相连，$1 \le i \le n$

约束函数B_i：新加入的结点i与之前结点能构成团，即有边相连

- (2) 设G的顶点数为5，判定是否存在大小为3的团，请给出该规模的解空间树
  - 叶节点共计32个。



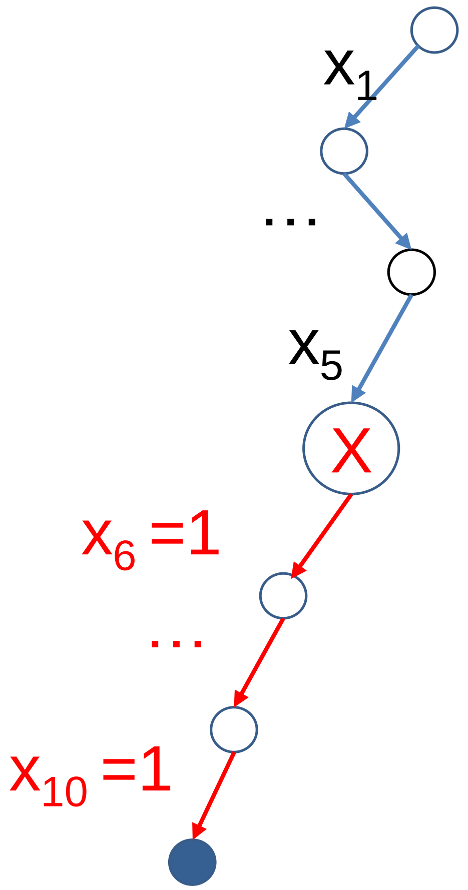
解空间树2^n个叶结点，判断每条路径的团O(n)，最坏情况下算法时间复杂度$O(n2^n)$

- 设计思想参考，设$x_1, \ldots, x_k$是根到X的路径：
- 1) 成本上界U
  - 初值：0
- 2) 成本上界函数u(X)

$\sum x_i$，i=1,..,k表示选中的顶点个数

  - $u(X)=-\sum x_i$，i=1,..,k
  - 当前若X是可行解，且$-\sum x_i<U$，i=1,..,k，则赋值给U。
- 3) 成本下界$\hat{c}(X)=f(X)+\hat{g}(X)$
  - $f(X)= -\sum x_i$，i=1,..,k
  - $\hat{g}(X)= -(n-k)$，最好情况下其余顶点都能加入到当前团中。
- 4) 约束函数B(X)
  - 任意两个x_i=1对应的顶点i之间都存在边
- 5) 基于FIFO-分支限界法或LC-分支限界法实现
  - 当$\hat{c}(X)$>U时，X舍弃

- 问题举例：设当前n=10，k=5，U=-2
- 算法生成结点X，X表示路径$(x_1,\ldots, x_5)=(1,0,0,1,1)$到达的状态结点
  - 1) 约束函数B(X)=T，即当前顶点1,4,5能够构成一个大小为3个团
  - 2) 计算ĉ(X)=-3-5=-8<U，加入活结点表中
  - 3) 计算u(X)=-3<U，更新U为-3
  - 4) 生成下一个活结点或选择下一个E结点



本 章 结 束