算法作业4：

6-17：

解：①：该问题等价于物品无限的背包问题

②：该动态规划问题需要填一个一维表，长度为 $v+1$

③：~~初始条件为：$F(0)=0$~~ 表中每个位置为 true / ~~false~~ false

true 表示此位置可被 $x_1 \cdots x_n$ 表示，false 则反之

④：此问题结果在 $F(v)$ 处取到。

⑤：动态规划范式为：$F(i)=\bigvee_{k=1,\cdots n} F(i-x_k)$（若 ~~…~~ $i<x_k$，$F(i-x_k)=false$）

⑥：时间复杂度分析：共有 $v+1$ 个位置需填，每个位置算 n 次

故为 $O(nv)$

6-20：

解：①：该问题是树状结构，并且是希望求解最优解，

在存在子树的条件下可考虑动态规划

②：因为是二叉搜索树并且子问题为子树，故需同时关注

某个树的 start_index 与 end_index，故为二维表，

维数为 $(n+1)\times(n+1)$

③：初值为：$F(i,i)=p_i$（$i\neq 0$），$F(i,j)=0$（即一个节点层数加或无节点），~~且只填~~ 只需填 $F(i,j)$，（$i<j$）的部分

④：最终结果在 $F(1,n)$ 处找到（行表示 start，列表示 end）

⑤：动态规划范式为：（上三角矩阵）。

~~cost（…）=~~

$$\begin{cases} cost(i,j)=\sum_{i\le k\le j} p_k + \min_{i\le k\le j}\left(cost(i,k-1),\ cost(k+1,j)\right) & (i<j) \\ cost(i,j)=0, & (i>j) \end{cases}$$

（~~即为~~ 其中 $cost(i,k-1)$ 为左子树 costs，$cost(k+1,j)$ 为右子树 costs，

$\sum p_k$ 是因为加上根 k 后，所有子树下沉一层，故多加一遍）

扫描全能王 创建

⑥：时间复杂度分析：

·共填 $(n+1)^2$ 个格子，每个格子至多算 n 次，即算 cost(i, n)

故为 $O(n^3)$.

6-21：（该问题等价于最大独立集问题）

解：①：与上一题类似，由于仍是树状结构，故子问题存在（即子树），故拟使用动态规划解决；

②：只关注顶点，故只填一个一维表，长度为 |V|+1

③：初值 F(0) = 0，

④：动态规划范式：

$$F(i) = \min\{1+\sum_{j为i孩子} F(j),\ \sum_{k为i孩子的孩子} F(k)\}.$$

⑤：时间复杂度分析：共有 |V| 个子问题，而所有子问题会遍历所有边且不重复（因为只会去找自己的孩子 node），故为 O(|V|+|E|)，为线性，符合条件.

6-22：

解：①：该问题类似于单一物品的二背包问题，可用线性规划求解

②：因为所有数只可用1次，故填二维表，行 i 表示使用 $\{a_1 \ldots, a_i\}$，列 j 表示整数 j，

故为一个 $(n+1)\times(t+1)$ 的表，且表中每个位置为1个 bool 值

③：F(0,0) = true，F(0,j)(j≠0) = false

F(i,0)(i≠0) = true，

④：范式有：$F(i,j) = \vee\{F(i-1, j) \vee F(i-1, j-a_i)\}$

且若 $j-a_i<0$，对应为 false

其中：$F(i-1,j)$ 表示不使用 a_i，$F(i-1,j-a_i)$ 表示使用了 a_i，

④：结果：检查 $F(k,t)$，$(k\in[1,i])$，若存在一个 k，使 $F(k,t)=true$ 则为 true，即：

$$ans=\bigvee_{k\in[1,i]} F(k,t)$$

⑤：时间复杂度分析：

填表需 $O(nt)$，而每一个格子 $O(1)$，最终结果获取需 $O(n)$，故总体为 $O(nt)$，满足要求

扫描全能王 创建