4.3.6.6 KP 的 dual PTAS

解法思路：使用 Bin-P

Step 1: 距离函数：

$$h(I,T) = \max\left\{\frac{\sum_{i\in T} w_i - b}{nb}, 0\right\}$$

Step 2: s-KP: $I = \{q_1, q_2, \dots, q_s, b, n_1, n_2, \dots, n_s, c_1, c_2, \dots, c_n\}$

Step 3: DP-KPs: $(m_1, m_2, \dots, m_s) \in \{0, \dots, n_1\} \times \{0, \dots, n_2\} \times \cdots \times \{0, \dots, n_2\}$

1. $DP(m_1, m_2, \dots, m_s, d) = \max\{DP(m_1, \dots, m_i - 1, \dots, m_s, d - q_i) + c_{i_1}\}$
2. 可以穷举所有的满足背包容量的选择，即 $\sum_{i=1}^{s} m_i q_i \le b$。然后计算它们的代价，$\forall i \le s$，$n_i$ 个物品中 m_i 个价值最大的被使用。

穷举的代价为 $O(n_1 \cdot n_2 \cdot \cdots \cdot n_s) \le O\left((n/s)^s\right)$

Step 4: KP-PTAϵ 和 KP-PTAS

令 $s = [1/\epsilon], l_i = 0$ 且 $l_i = (i-1)\epsilon b, l_{s+1} = b$

通过 rounding 得到 $I' = \{l_1, l_2, \dots, l_n, b, n_1, n_2, \dots, n_s, c_1, c_2, \dots, c_n\}$，对 I' 使用 DP-KPs

1. 时间复杂度 $O\left((n\epsilon)^{1/\epsilon}\right)$
2. $cost(KP - PTAS(I, \epsilon)) = cost(DP - KPs(I')) \ge Opt_{KP}(I)$，因为物品的重量减少了（现在能装入更多的物品）
3. $\forall i \in T, w_i - w_i' \le \epsilon b$ 且 $\sum_{i\in T} w_i' \le b$

$$h(I,T) = \max\left\{\frac{\sum_{i\in T} w_i - b}{nb}, 0\right\} \le \frac{\sum_{i\in T}(w_i' + \epsilon b) - b}{nb} \le \frac{b + n\epsilon b - b}{nb} = \epsilon$$

5.3.3.9

(i)

$$Jac\left[\frac{ab}{n}\right] = \prod_{i=1}^{l}\left((ab)^{(p_i-1)/2} \mod p_i\right) = \prod_{i=1}^{l}\left(a^{(p_i-1)/2} \mod p_i\right)\prod_{i=1}^{l}\left(b^{(p_i-1)/2} \mod p_i\right) = Jac\left[\frac{a}{n}\right]Jac\left[\frac{b}{n}\right]$$

(ii) 若 $a \equiv b \mod p$，则 $a \equiv b \mod q$，其中 q 是 p 的某个因数：

$$Jac\left[\frac{a}{n}\right] = \prod_{i=1}^{l}\left(a^{(p_i-1)/2} \mod p_i\right) = \prod_{i=1}^{l}\left(b^{(p_i-1)/2} \mod p_i\right) = Jac\left[\frac{b}{n}\right]$$

(iii) 本题的重点是需要用到”quadratic reciprocity”，使得 $Leg[p/q]$ 和 $Leg[q/p]$ 能够建立关系。

令 $n = p_1^{k_1} p_2^{k_2} \cdots p_l^{k_l}, a = q_1^{r_1} q_2^{r_2} \cdots q_m^{r_m}$，其中 $p_1, p_2, \cdots, p_l, q_1, q_2, \cdots, q_m$ 为成对的不同奇质数，由 Jac 和 Leg 符号的定义，可得：

$$Jac\left[\frac{a}{n}\right] = \prod_{i=1}^{l}\left(Leg\left[\frac{a}{p_i}\right]\right)^{k_i} = \prod_{i=1}^{l}\prod_{j=1}^{m}\left(Leg\left[\frac{q_j}{p_i}\right]\right)^{k_i r_j}$$

再由 *law of quadratic reciprocity*, 即 $Leg[p/q]Leg[q/p] = (-1)^{\frac{p-1}{2}\frac{q-1}{2}}$，其中 p,q 为不同的奇质数，由二项式定理：

$$
\begin{aligned}
\frac{n-1}{2} &= \frac{\prod_{i=1}^{l} p_i^{k_l} - 1}{2} \\
&= \frac{\prod_{i=1}^{l}\left(2\cdot\frac{p_i-1}{2}+1\right)^{k_l} - 1}{2} \\
&\equiv \frac{\prod_{i=1}^{l}\left(2k_i\cdot\frac{p_i-1}{2}+1\right) - 1}{2} \mod 2 \\
&\equiv \sum_{i=1}^{l} k_i\left(\frac{p_i-1}{2}\right) \mod 2 \\
\therefore Jac\left[\frac{a}{n}\right] &= \prod_{i=1}^{l}\prod_{j=1}^{m}\left(Lef\left[\frac{q_j}{p_i}\right]\right)^{k_i r_j} \\
&= \prod_{i=1}^{l}\prod_{j=1}^{m}\left(Lef\left[\frac{q_j}{p_i}\right]\right)^{k_i r_j}(-1)^{k_i\frac{p_i-1}{2}r_j\frac{q_i-1}{2}} \\
&= (-1)^{\left(\sum_{i=1}^{l}k_i\frac{p_i-1}{2}\right)\left(\sum_{j=1}^{m}r_i\frac{q_i-1}{2}\right)}\prod_{i=1}^{l}\prod_{j=1}^{m}\left(Leg\left[\frac{p_i}{q_j}\right]\right)^{k_i r_j} \\
&= (-1)^{\frac{a-1}{2}\frac{n-1}{2}} Jac\left[\frac{n}{a}\right]
\end{aligned}
$$

(iv) 显然

$$Jac\left[\frac{1}{n}\right] = \prod_{i=1}^{l}\left(Leg\left[\frac{1}{p_i}\right]\right)^{k_1} = 1$$

(v) 使用 quadratic reciprocity 的第二补式：

$$Leg\left[\frac{2}{p}\right] = (-1)^{\frac{p^2-1}{8}} = \begin{cases} -1, & p \equiv 3,5 \mod 8 \\ 1, & p \equiv 1,7 \mod 8 \end{cases}$$

由 *Jac* 符号的定义：

$$Jac\left[\frac{2}{n}\right] = \prod_{i=1}^{l}\left(Leg\left[\frac{2}{p_i}\right]\right)^{k_i} = \prod_{i=1}^{l}\left((-1)^{\frac{p_i^2-1}{8}}\right)^{k_i}$$

再次使用二项式定理：

$$
\begin{aligned}
\frac{n^2-1}{8} &= \left(\left(\prod_{i=1}^{l} p_i^{k_i}\right)^2 - 1\right)\Big/8 \\
&= \frac{\prod_{i=1}^{l}\left(8\frac{p_i^2-1}{8}+1\right)^{k_i} - 1}{8} \\
&\equiv \frac{\prod_{i=1}^{l}\left(8k_i\frac{p_i^2-1}{8}+1\right) - 1}{8} \mod 2 \\
&\equiv \sum_{i=1}^{l} k_i\frac{p_i^2-1}{8} \mod 2 \\
\therefore Jac\left[\frac{2}{n}\right] &= \prod_{i=1}^{l}\left(Leg\left[\frac{2}{p_i}\right]\right)^{k_i} = \prod_{i=1}^{l}\left((-1)^{\frac{p_i^2-1}{8}}\right)^{k_i}
\end{aligned}
$$

$$= (-1)^{\sum_{i=1}^{l} k_i \frac{p_i^2-1}{8}} = (-1)^{\frac{n^2-1}{8}} = \begin{cases} -1, & n \equiv 3,5 \mod 8 \\ 1, & n \equiv 1,7 \mod 8 \end{cases}$$

5.3.3.10

Jacob(a, n) // 假定 a 和 n 互质
if a is even // (v),(i)
    if $n \equiv 3,5 \mod 8$
        return −Jacob(a/2,n)
    else
        return Jacob(a/2,n)
if $a == 1$
    return 1 // (iv)
if $(a \;\&\; 0x2 == 1)$ && $(b \;\&\; 0x2 == 1)$
    return -Jacob$(n \mod a, a)$
else
    return Jacob$(n \mod a, a)$

3.3.2.9

Set cover problem. $Pot(X)$ is the set of all subsets of the set X. Similar to VCP, use divide-and-conquer. Divide: $((X_i, F_i), k-1)$: Select any $S_i \in F$. Let $X_i = X \backslash S_i, F_i = f(F, i)$, function f deletes S_i and all elements of S_i in $S_1, \cdots, S_{i-1}, S_{i+1}, \cdots, S_l$. Using induction, $((X_T, F_T), 0)$ is trivial. Complexity is $O(Pat^{pat}|X|)$.

不必限制输入的网络流算法

改变算法 3.2.3.10 的第 3 步和第 4 步：

1. *Edmonds-Karp algorithm*: 通过 BFS 找到最短路
2. *Dinic's blocking flow algorithm*: 在 residual graph 上使用 BFS 构造一个分层图

branching-and-bound 解背包问题

Knapsack Problem: KP$(b, \{w_1, w_2, \cdots, w_n\}, \{c_1, c_2, \cdots, c_n\})$

(1) 建立一个 backtracking tree. 在树的每个内部节点，根据物品是否放入背包来选择分支

(2) 剪枝和 BFS/DFS，设当前可行解的价值 $cost = c$

   i. $\sum w_{k_i} > b$

   ii. 当前的价值为 a，当前的总重量 $weight = d$，当前还能放入的且性价比 (c_i/w_i) 最高的物品为 q，若 $a + (b-d) \cdot q < c$，则剪枝

(3) 有利的输入：最开始就找到了最优解，同样的剪枝策略被不断使用；不利输入：难以剪枝

3.4.2.1

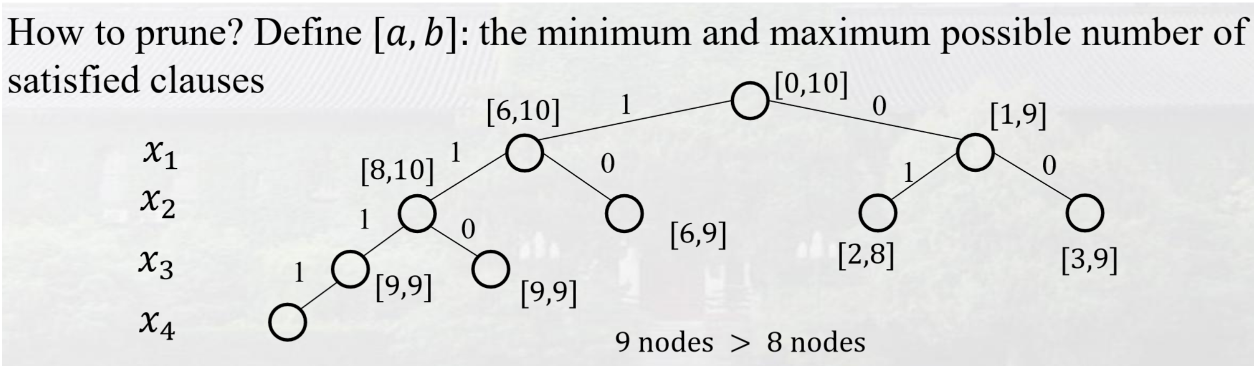

图 1: Ex 3.4.2.1 中的搜索树

3.4.2.2

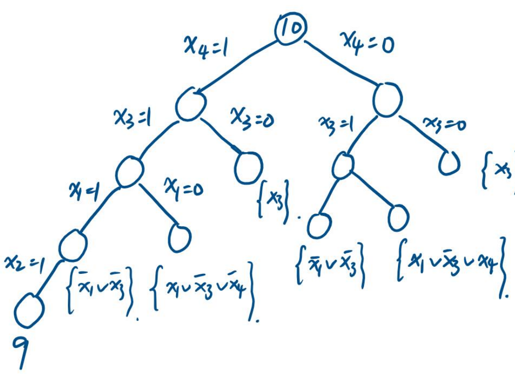

图 2: 右 DFS

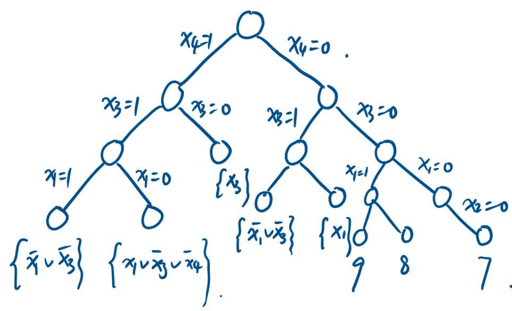

图 3: 左 DFS