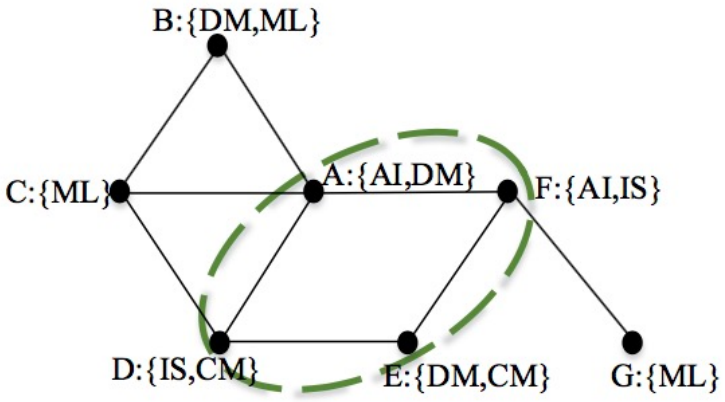
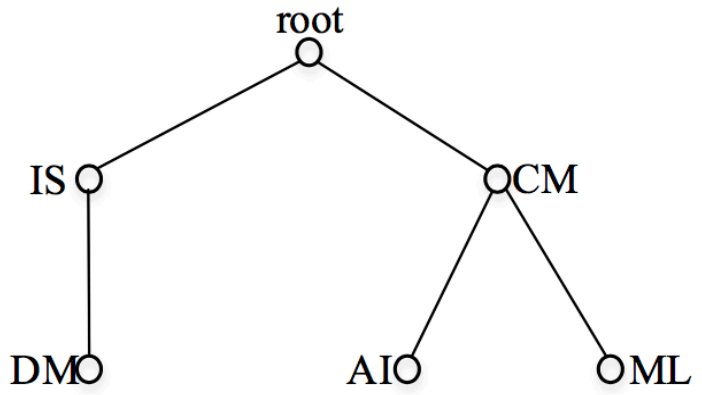
Hardness and compressed index

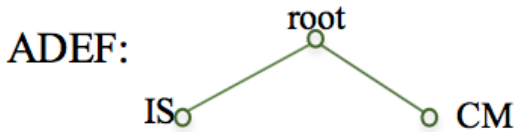
6.26

Problem definition

- Given a profiled graph G, a profile tree T, and one query node $q \in$ G, find communities, each of which is a subgraph G_q satisfying the following properties:
  - Connectivity: G_q is connected and $G_q \subseteq$ G;
  - Structure cohesiveness: k-core;
  - Semantic cohesiveness: all the vertices in G_q share the maximal common induced rooted subtree of q;







Transformation from tree to sequence

- Semantic cohesiveness: all the vertices in G_q share the maximal common induced rooted subtree of q;

- Mining maximal common subtree $\longleftrightarrow$ mining maximal common subsequence $\longleftrightarrow$ mining maximal frequent subsequence:

  (1) min_sup=k+1

  (2)subsequence of query vertex's.

Hardness of mining all frequent subsequences

- “If a counting problem is #P-complete[3] or #P-hard, then its associated problem of enumerating all solution must be NP-hard.”[1,2]

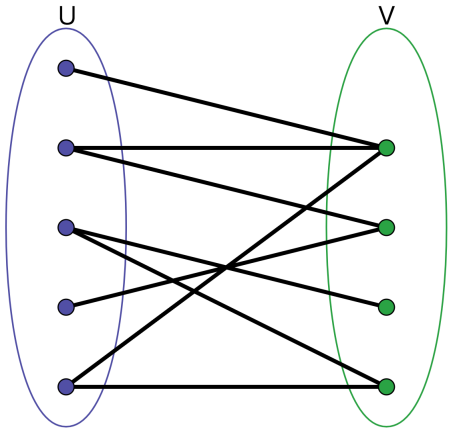
- Our target is to prove that counting the number of all maximal common subsequences is #P-hard.

[1].Michael R G, David S J. Computers and intractability: a guide to the theory of NP-completeness[J]. WH Free. Co., San Fr, 1979: 90-91.
[2].Papadimitriou C H. Computational complexity[M]. John Wiley and Sons Ltd., 2003.
[3].Valiant L G. The complexity of computing the permanent[J]. Theoretical computer science, 1979, 8(2): 189-201.

Bipartite graph

- $G = \{U, V, E\}$, U, V are two disjoint vertex sets, and E represent edges between two vertices in U and V.
- Bipartite clique: $G = \{U, V, E\}$. If there is an edge

Between every pair of vertices in U and V. G is bipartite clique.



- Maximal bipartite clique.

- "The problem of counting the number of maximal

Bipartite cliques in a given bipartite graph is #P-complete".[1]

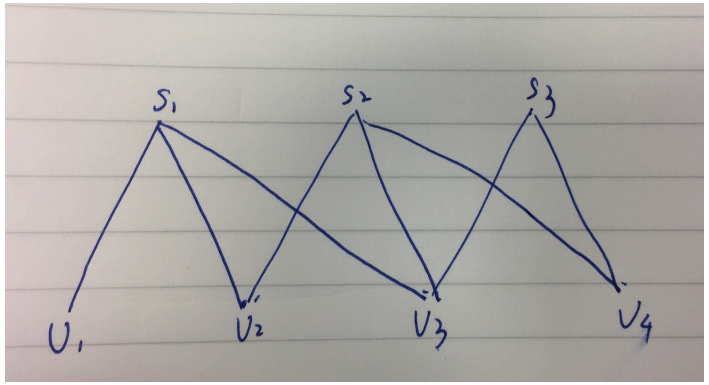
[1]Provan J S, Ball M O. The complexity of counting cuts and of computing the probability that a graph is connected[J]. SIAM Journal on Computing, 1983, 12(4): 777-788.

Cont.

- Let $S = \{s_1, s_2, \cdots, s_n\}$ represents all items of P-trees. $U = \{u_1, u_2, \cdots, u_n\}$ represents all users in G.

- mining maximal subsequences $\longleftrightarrow$

  mining all maximal bipartite cliques.



- However, “common”(frequency) has not been included.

Cont.

- Let $\mathrm{F}_i(D)$ denote all maximal frequent subsequences of D whose support is i.
- Let $M_i(D)$ denote all maximal frequent subsequences of D whose support is no less than i.

- mining maximal subsequences $\Longleftrightarrow$ mining all maximal bipartite cliques.
- all maximal bipartite cliques are $\sum_{i=0}^{max} \mathrm{F}_i(D)$.

- Corollary: The problem of counting the number of $\sum_{i=0}^{max} |\mathrm{F}_i(D)|$ is #P-complete.

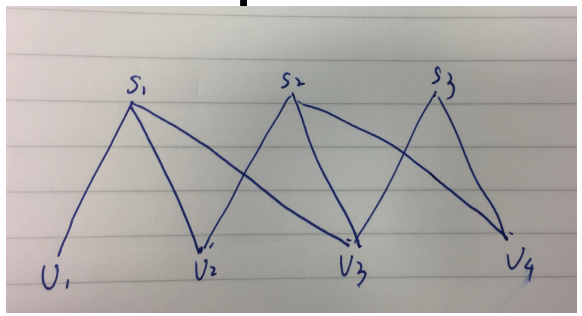
Cont.

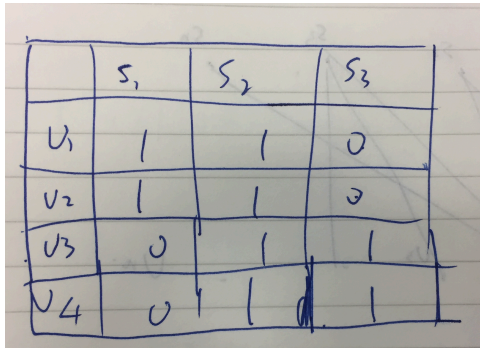
- Lemma: It is a #P-hard to count the number of all maximal common subsequences.

- Proposition: $\mathrm{F}_k(D) \subseteq M_k(D) \subseteq \sum_{i=k}^{max} \mathrm{F}_i(D)$.

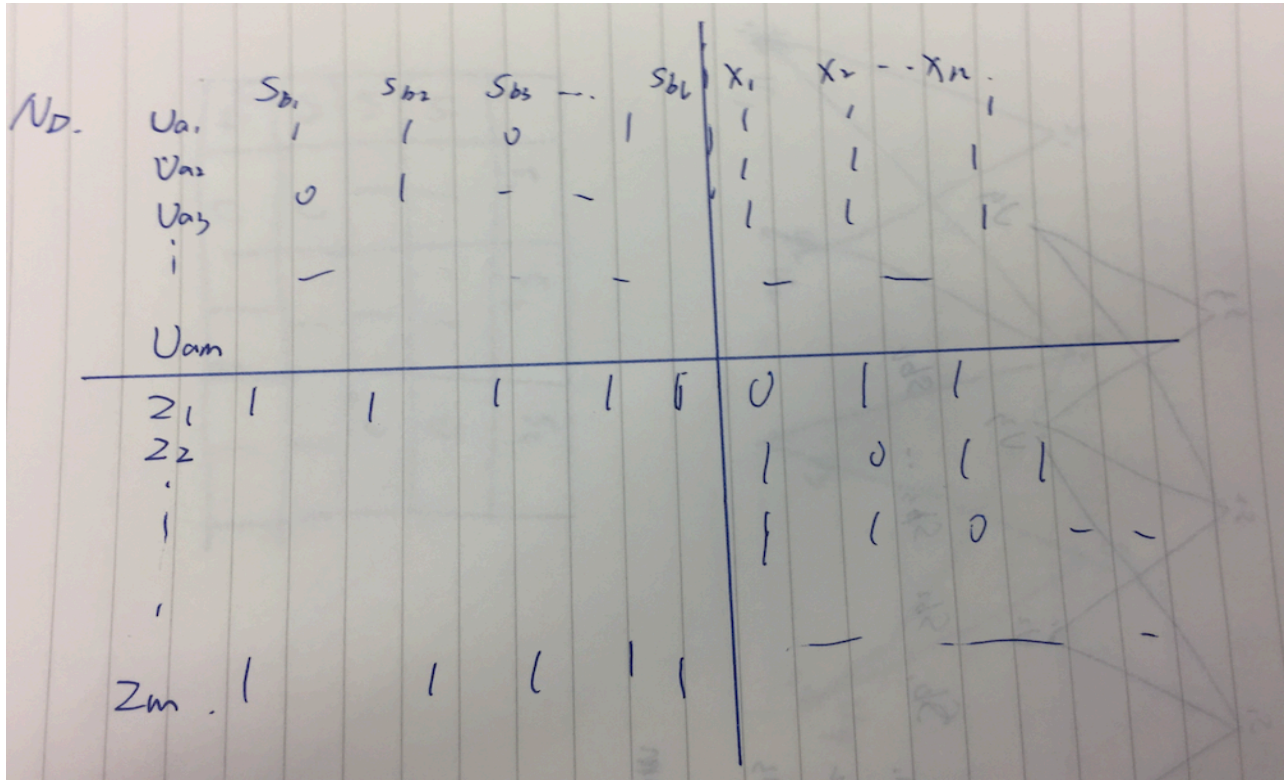
- Our target is to create a new database N_D and $|M_k(N_D)|$ has equation with $\sum_{i=k}^{max} \mathrm{F}_i(D)$. Then we can reduce *Corollary* to lemma and prove counting the number of all maximal common subsequences is at least hard as *Corollary.*

Matrix representation



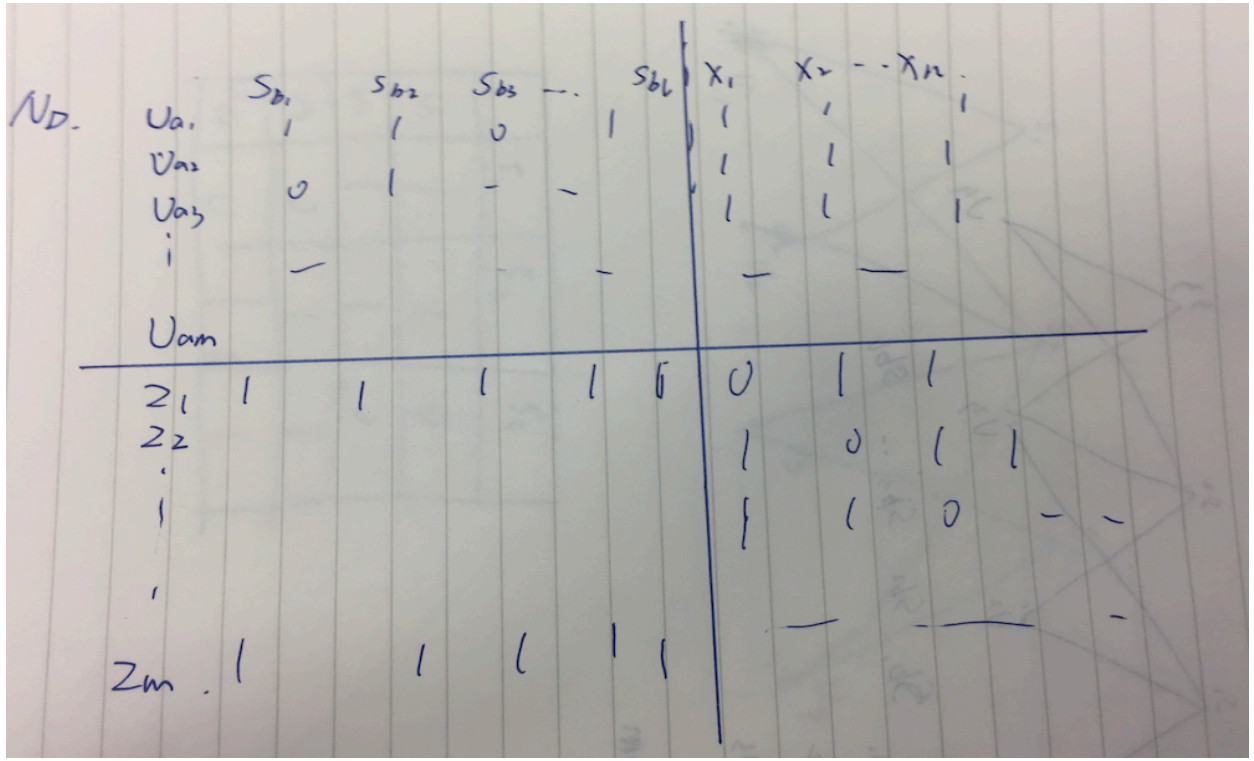


- $N_D\ upper = D_q$ induced by query Vertex+ X;
- $N_D\ lower = S_Q + X - \{X_i\}$;



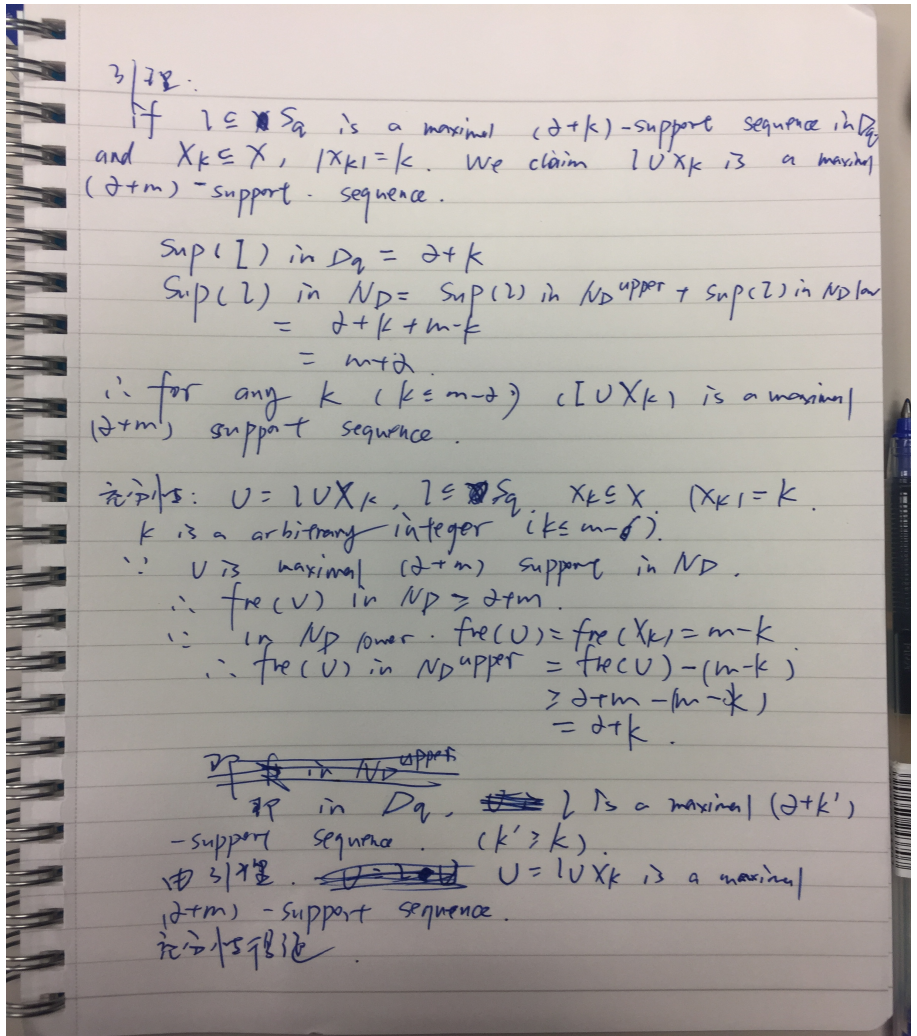
- Lemma: If I $\subseteq S_q$ is a maximal (a+k)-support sequence in D_q and $X_k \subseteq X$. Then I $\cup X_k$ is a maximal (a+m)-support sequence in N_D

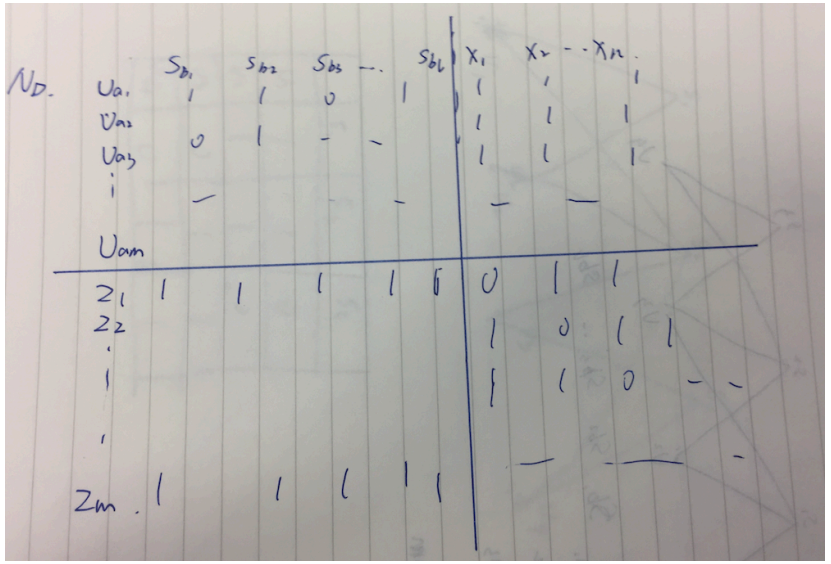
- Theorem: sequence U is a maximal (a+m) common subsequence in N_D iff U's support is (a+m) in N_D.



proof

- Theorem: sequence U is a maximal $(a+\mathrm{m})$ common subsequence in N_D iff U's support is $(a+\mathrm{m})$ in N_D.
- ⟵ from definition.
- ⟶





Conclusion

- Theorem: sequence U is a maximal (σ+m) common subsequence in N_D iff U's support is (σ+m) in N_D.

$$|M_{a+m}(N_D)| = |F_{a+m}(N_D)|$$

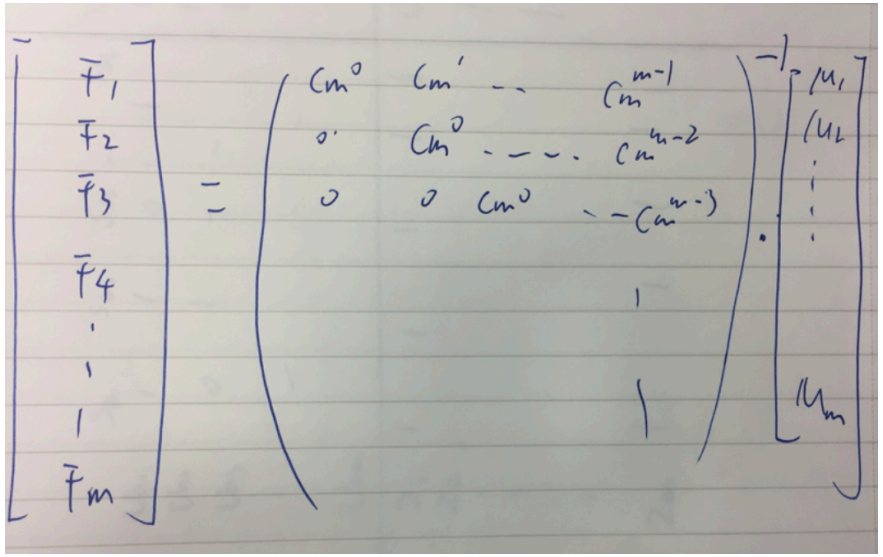
Lemma: If I $\subseteq S_q$ is a maximal (a+k)-support sequence in D_q and $X_k \subseteq X$. Then I $\cup$ X_k is a maximal (a+m)-support sequence in N_D.

$$|F_{a+m}(N_D)| = \sum_{i=0}^{m-a} |\mathrm{F}_{a+i}(D)| \cdot C_m^i.$$

Then: $|M_{a+m}(N_D)| = \sum_{i=0}^{m-a} |\mathrm{F}_{a+i}(D)| \cdot C_m^i$.

Reduction

- $|M_{a+m}(N_D)| = \sum_{i=0}^{m-a} |\mathrm{F}_{a+i}(D)| \cdot C_m^i$.



- If there exists a polynomial-time algorithm to count the number of maximal common subsequences, then $|M_{a+m}(N_D)|$ can be computed in polynomial time, then (F1,F2,…,Fm) can be computed in polynomial time.
- Corollary: The problem of counting the number of $\sum_{i=0}^{max} |\mathrm{F}_i(D)|$ is #P-complete.
- Then <u>Lemma: It is a #P-complete to count the number of all maximal common subsequences.</u>

compressed index-mathematic preliminary

- The only divisors of a prime integer p (p>1) are 1 and p.
- Every positive integer n is either 1 or can be expressed as a product of several prime integers, and this factorization is unique with the order of prime integers. The *standard form* of n factorization of n: $n=p_1{}^{m_1}p_2{}^{m_2}\cdots p_n{}^{m_n}$, p_i is a distinct prime integer, m_i is called the *multiplicity* of p_i.

- Give two integers a and b, the great common divisor of a, b is $\gcd(a,b)$. E.g., a =$2^3\cdot 3^2\cdot 7 = 504$, $b = 2^2\cdot 3^1\cdot 7\cdot 11$ =924, $\gcd(a,b) = 2^2\cdot 3^1\cdot 7 = 84$.

- if we set m_i as 1. Then $n=p_1p_2\cdots p_n$. $\gcd(a,b)=\prod_{i=1}^{m} p_{x_i}$.

compressed index-example

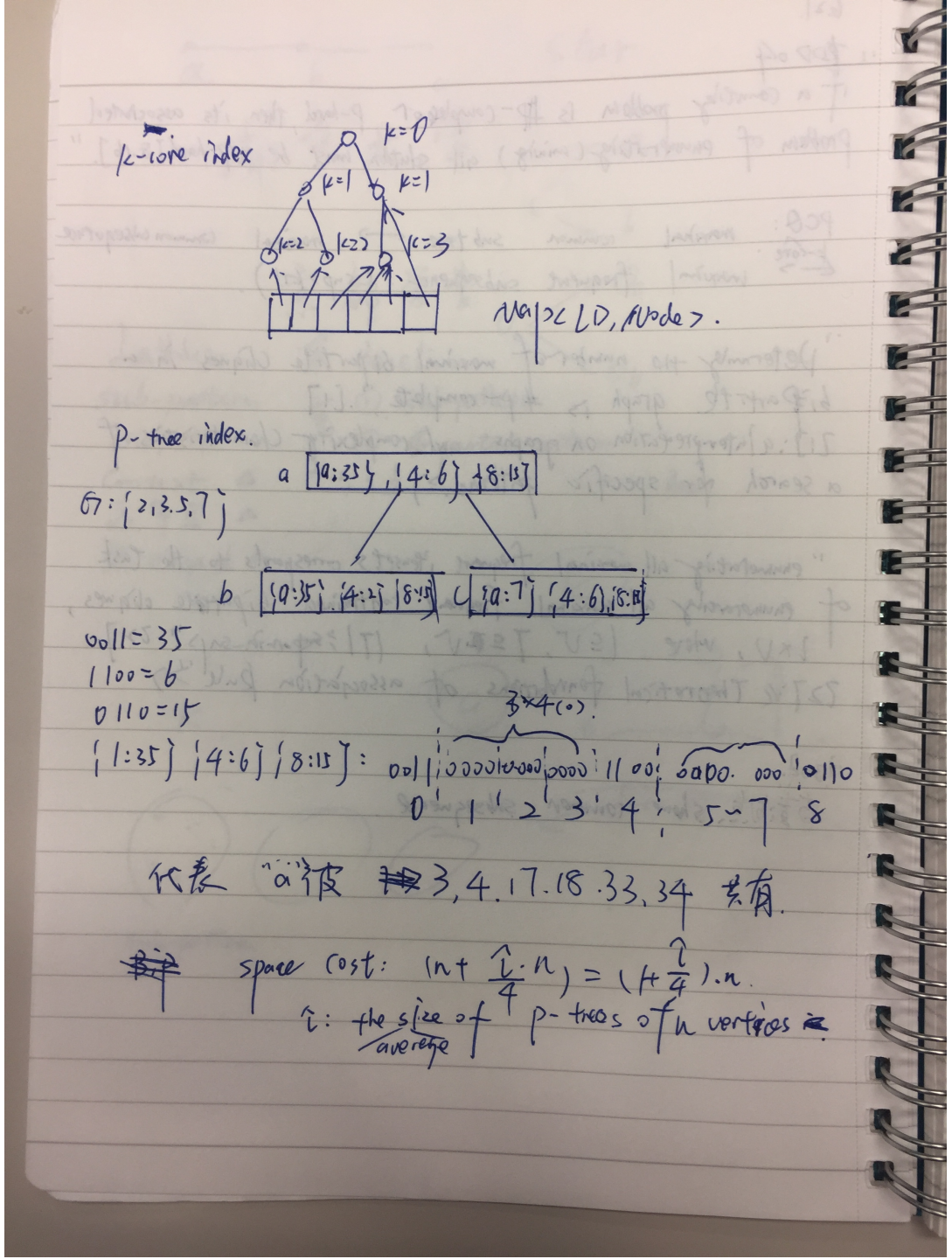
- $G = \{2,3,5,7\}$, $a = 2 \cdot 3 = 6$, $b = 5 \cdot 7$=35, gcd(a,b)=1.
- $S_a = 1100$, $S_b = 0011$, $S_a \cap S_b$=0000=1.

- Set 4 bits as a block.S_a={10 ,$\cdots$}.

Then $S_a \cap S_b$={gcd(S_{a_i}, S_{b_i})| $i \in number\ of\ blocks$}.

- If G={2,3,5,7}, then gcd(a,b) has $2^4 \cdot 2^4 \cdot 0.5$=128 types which can be pre-computed and stored in a table.

Index 3


k-core index
k=0
k=1
k=1
k=2
k=2
k=3
<vid, LD, Node>.
p-tree index.
a
b
c
0011=35
1100=6
0110=15
3×4(=).
0 1 2 3 4 5~7 8
代表 "a"的 3,4,17,18,33,34 共有.
space cost: (n + î/4 · n) = (1 + î/4) · n.
î: the size of p-trees of n vertices in