Parameterized Complexity of Biclique Cover and Partition

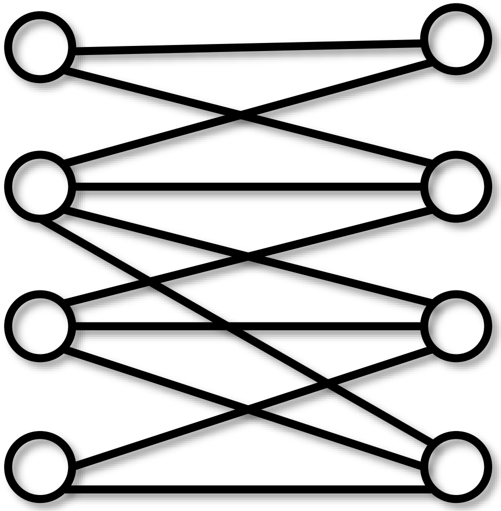
Davis Issac

Charles University, Prague

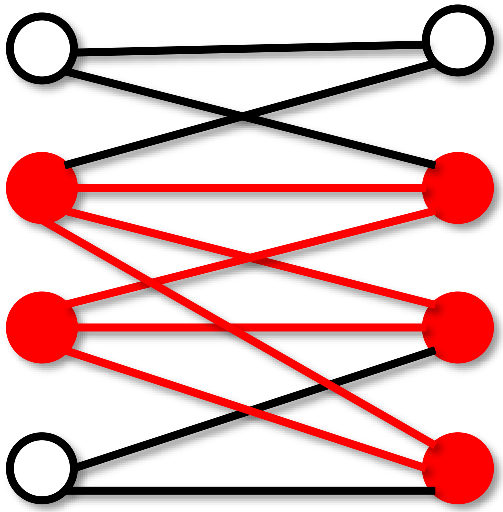
Joint work with

Sunil Chandran @ Indian Institute of Science, Bangalore
Andreas Karrenbauer @ MPI Saarbruecken, Germany

Input: Bipartite Graph

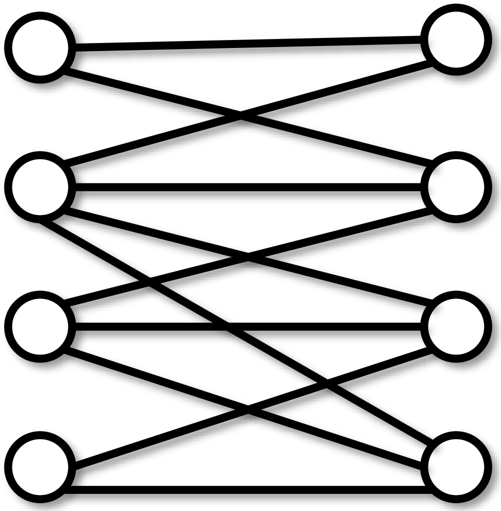


Biclique



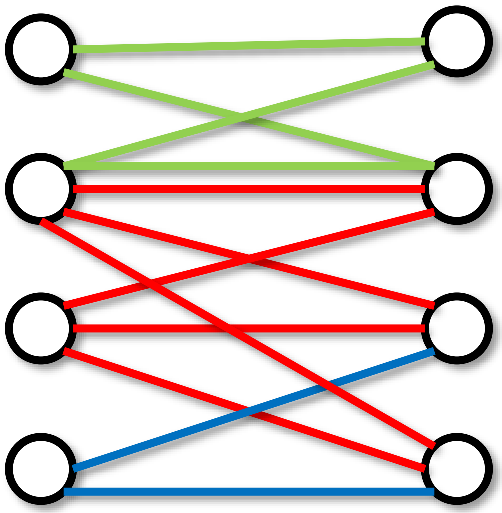
Complete Bipartite Subgraph

Biclique Cover



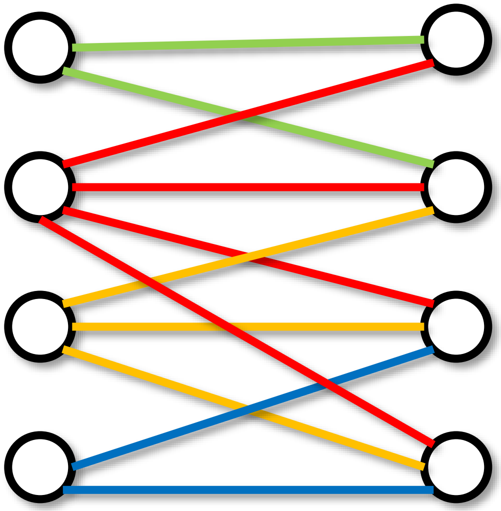
Given graph G, can the edges of G be covered by k bicliques ?

Biclique Cover



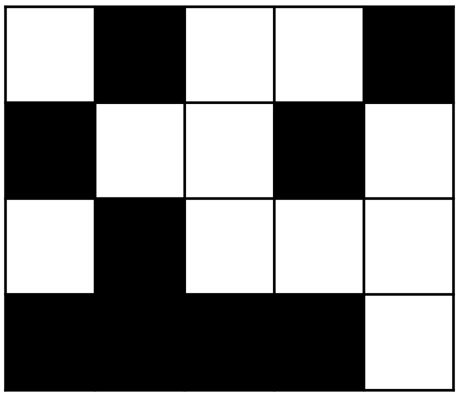
Given graph G, can the edges of G be covered by k bicliques ?

Biclique Partition

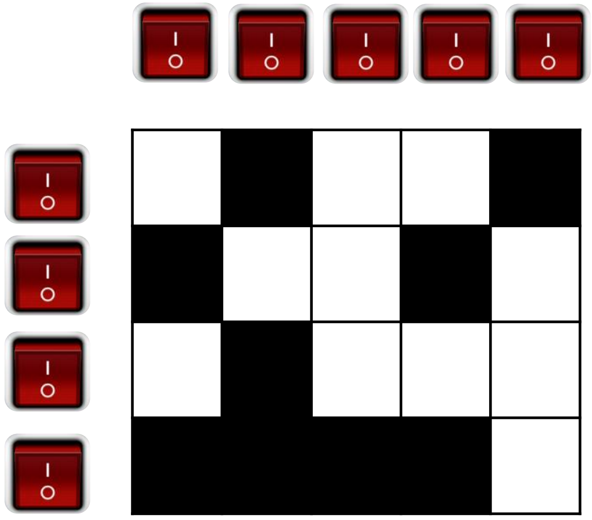


Given graph G, can the edges of G be partitioned into k bicliques ?

Application in Display Optimization

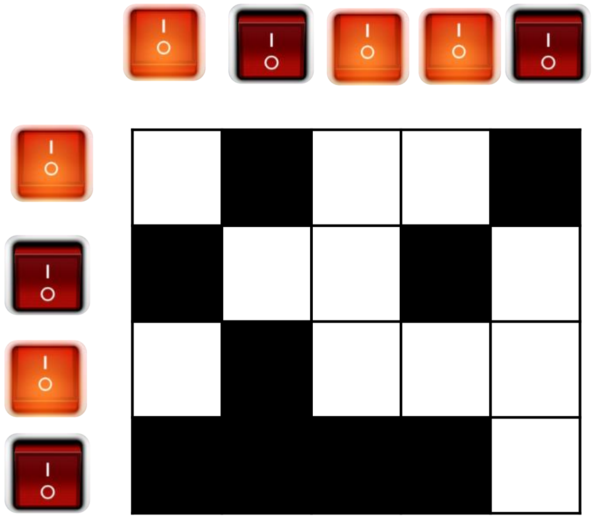


Application in Display Optimization

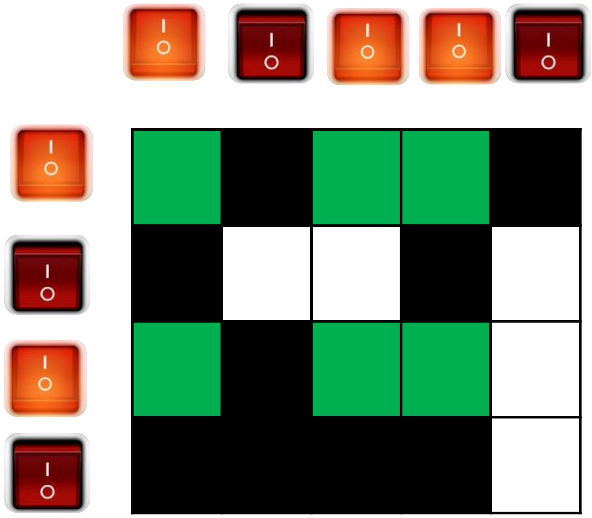


Passive Matrix OLED displays

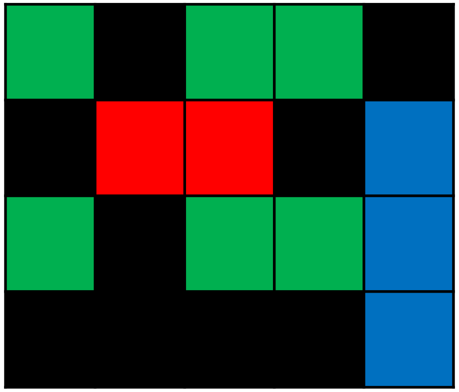
Application in Display Optimization



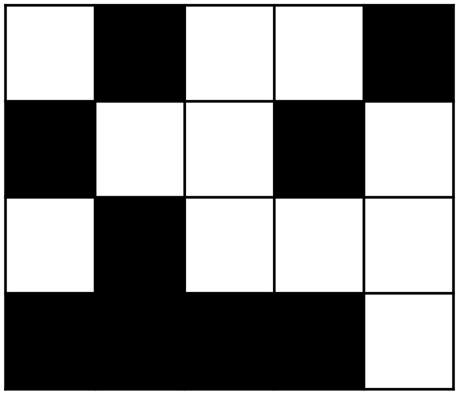
Application in Display Optimization

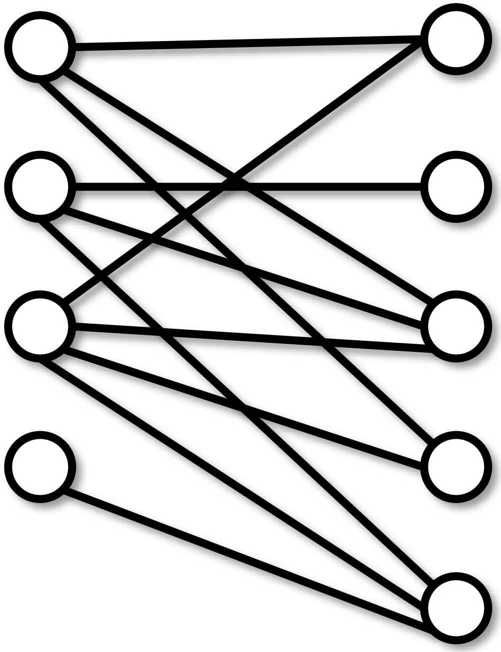


Application in Display Optimization

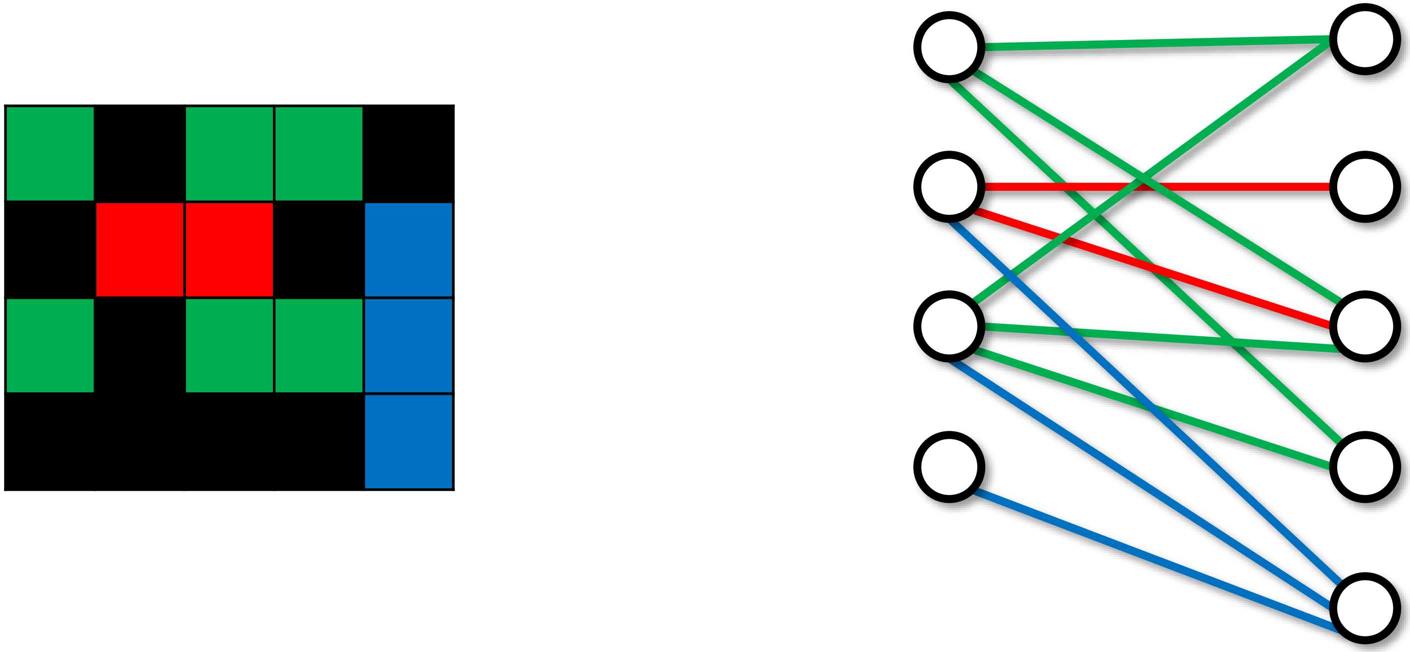


Application in Display Optimization



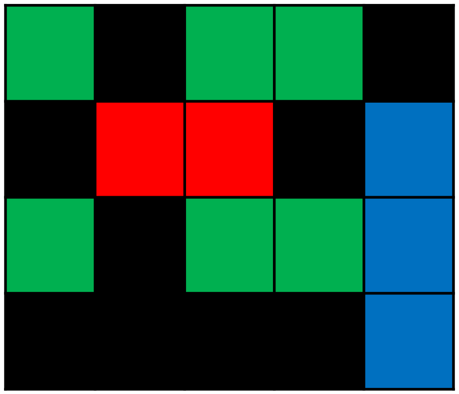


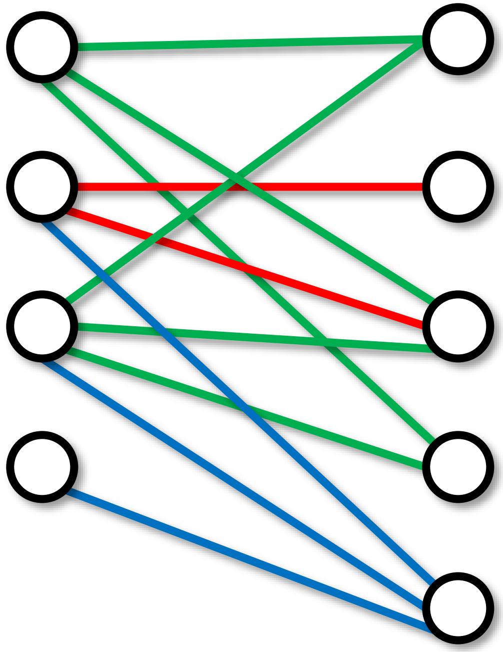
Application in Display Optimization



The addressing problem maps to Biclique partition

Application in Display Optimization

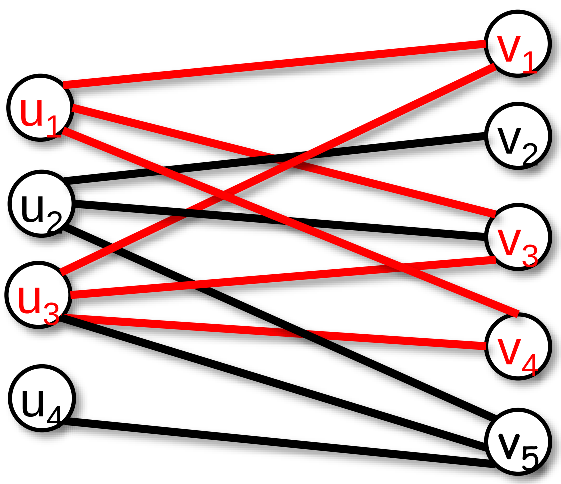




In Plasma displays, where each pixel has a memory, the addressing problem maps to Biclique cover.

Biclique in Matrix terms

Bicliques correspond to Rank-1 sub-matrices



	v1	v2	v3	v4	v5
u1	1	0	1	1	0
u2	0	0	0	0	0
u3	1	0	1	1	0
u4	0	0	0	0	0

Equivalence with Matrix ranks

Biclique Cover is equivalent to Boolean Rank

The minimum number of binary rank-1 matrices whose Boolean sum is equal to the given matrix

$$\begin{matrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{matrix} = \begin{matrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{matrix} + \begin{matrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{matrix} + \begin{matrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{matrix}$$

Equivalence with Matrix ranks

Biclique Partition is equivalent to Binary Rank

The minimum number of binary rank-1 matrices whose standard sum is equal to the given matrix

1	0	1	1	0
0	1	1	0	1
1	1	1	1	1
0	0	0	0	1

=

1	0	1	1	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

+

0	0	0	0	0
0	1	1	0	1
0	0	0	0	0
0	0	0	0	0

+

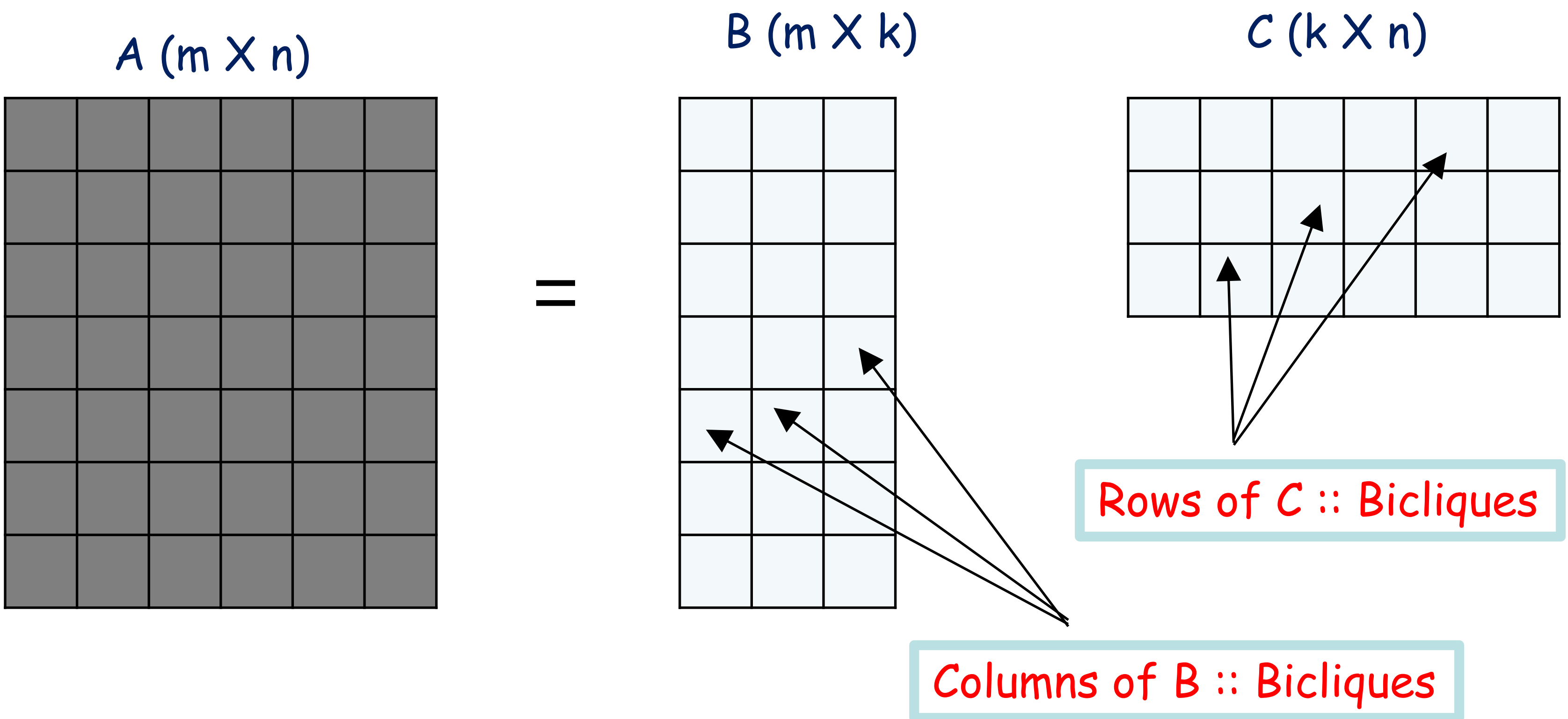
0	0	0	0	0
0	0	0	0	0
1	1	1	1	1
0	0	0	0	0

+

0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	1

Equivalent Definition of Boolean Rank of a matrix

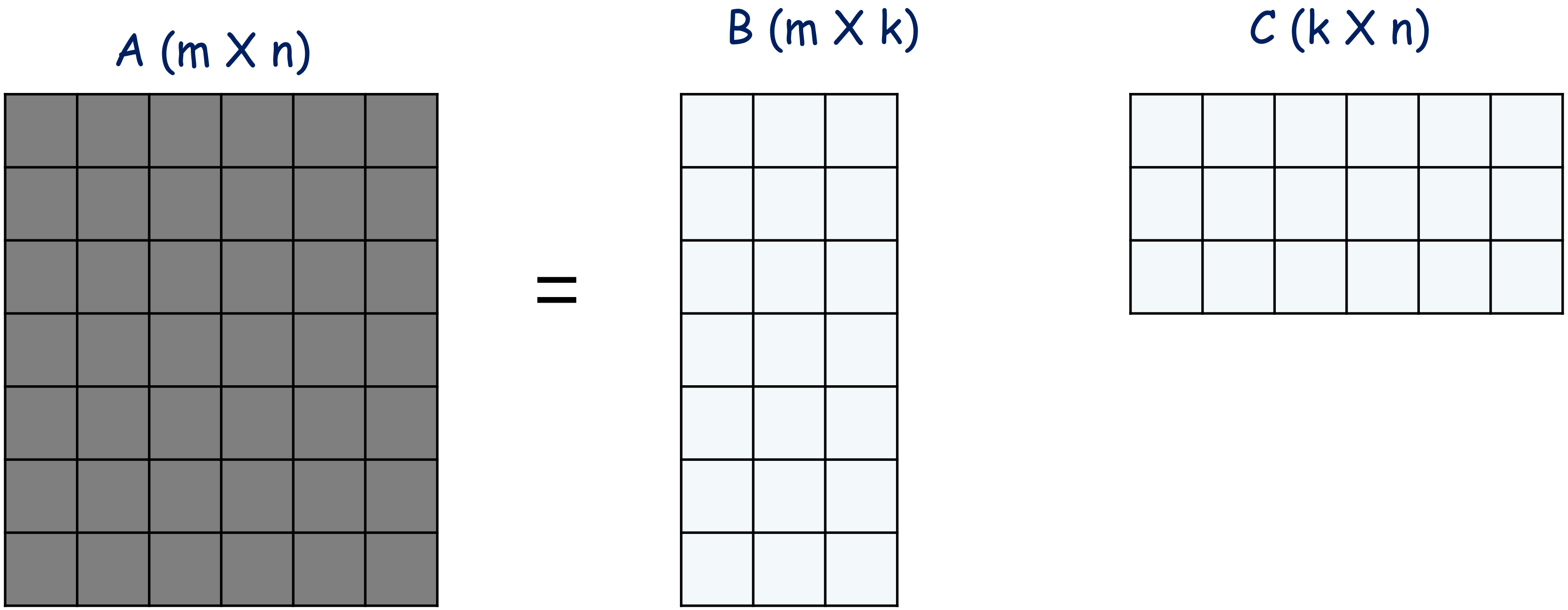
Smallest k such that



1. B and C are binary matrices
2. product is over Boolean arithmetic (1+1=1)

Equivalent definition of Binary Rank

Smallest k such that



1. B and C are binary matrices
2. product is over standard arithmetic (1+1=2)

Application in Data Compression, Mining

- Representing A by B, C takes only (m+n)k space instead of mn
- Matrix-vector product Ax takes (m+n)k time instead of mn
- Many real-world data is binary
- It is desirable to have the compressed matrices also as binary
- Boolean Matrix factorization has been used in data mining applications

Other Applications

- Bioinformatics
- Graph Drawing
- Automata Minimization
- Communication Complexity
-
-

NP-hardness

- Bipartite Biclique Cover is NP-hard [Orlin 77]
- Bipartite Biclique Partition is NP-hard [Jiang, Ravikumar 93]

Parameterized complexity

- Biclique Cover and Partition admit a kernel of size 2^{k+1} [Fleischner, Mujuni, Paulusma '07]

Kernel for bipartite graphs

- Reduction rule: If there are twin vertices, delete one of them
- Suppose there are no twins
- Let $B_1, B_2, \ldots, B_k$ be a cover/partition
- Let $S_v = \{i : v \in B_i\}$
- $S_u \neq S_v$
- Number of vertices is at most 2^k in each bipartition

Parameterized Complexity

- Biclique Cover and Partition admit a kernel of size 2^{k+1} [Fleischner, Mujuni, Paulusma, Szeider '09]
- Implies that Biclique Cover and Partition admits FPT algorithms with runtime $O^*(2^{(2^{2k} \log k)})$

[Chandran, I., Karrenbauer IPEC 2016]

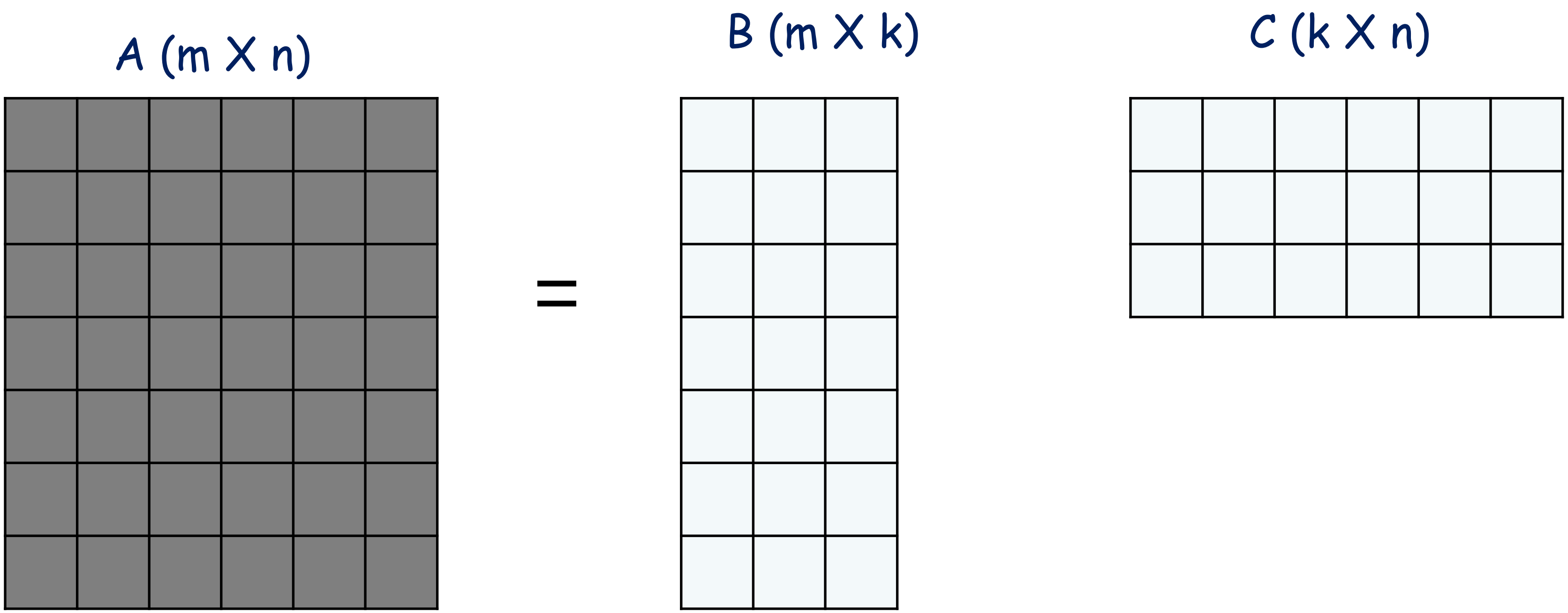
1. Biclique Partition has an algorithm running in time $2^{(2k^2+k+\log k)} + \mathrm{O}(\mathrm{n}_1\mathrm{n}_2(\log \mathrm{n}_1 + \log \mathrm{n}_2))$
2. Biclique Cover has no algorithm running in $2^{2^{o(k)}}\mathrm{poly(n)}$ time
3. Biclique Cover has no kernel of size $2^{o(k)}$

Algorithm for Biclique Partition

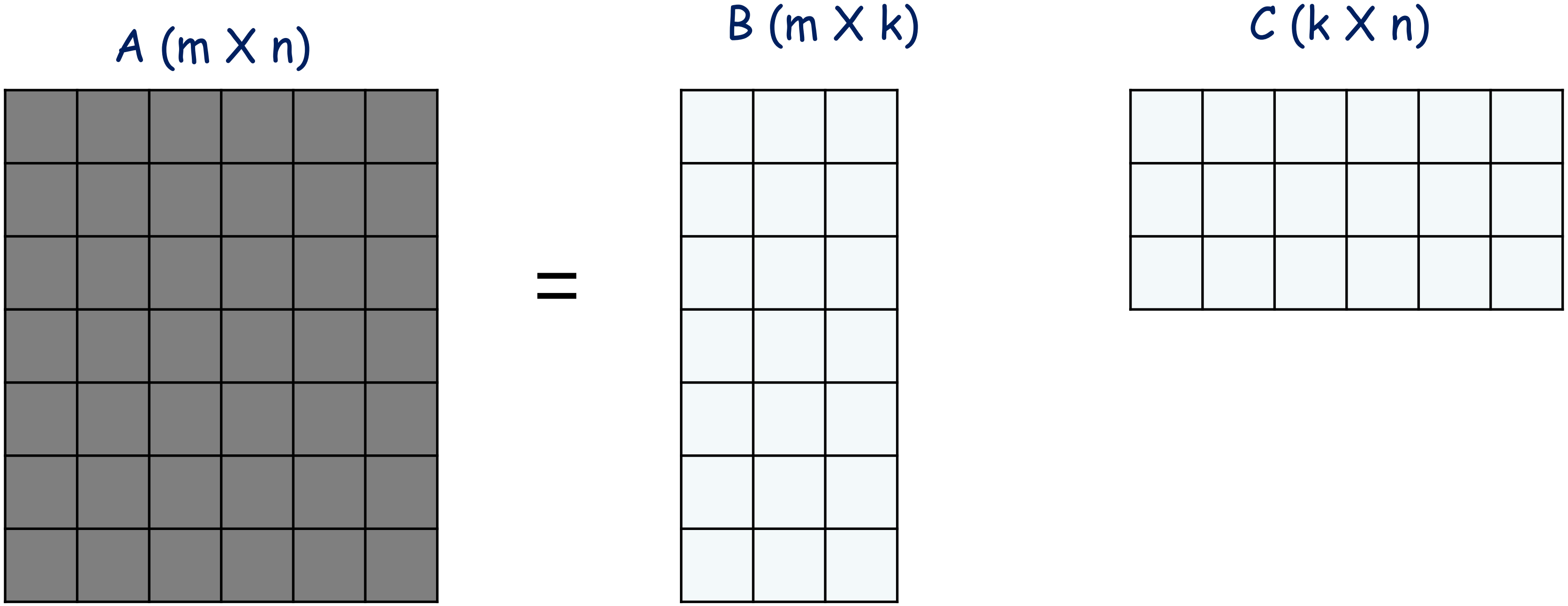
We will give an algorithm for k-Binary rank

k-Binary Rank

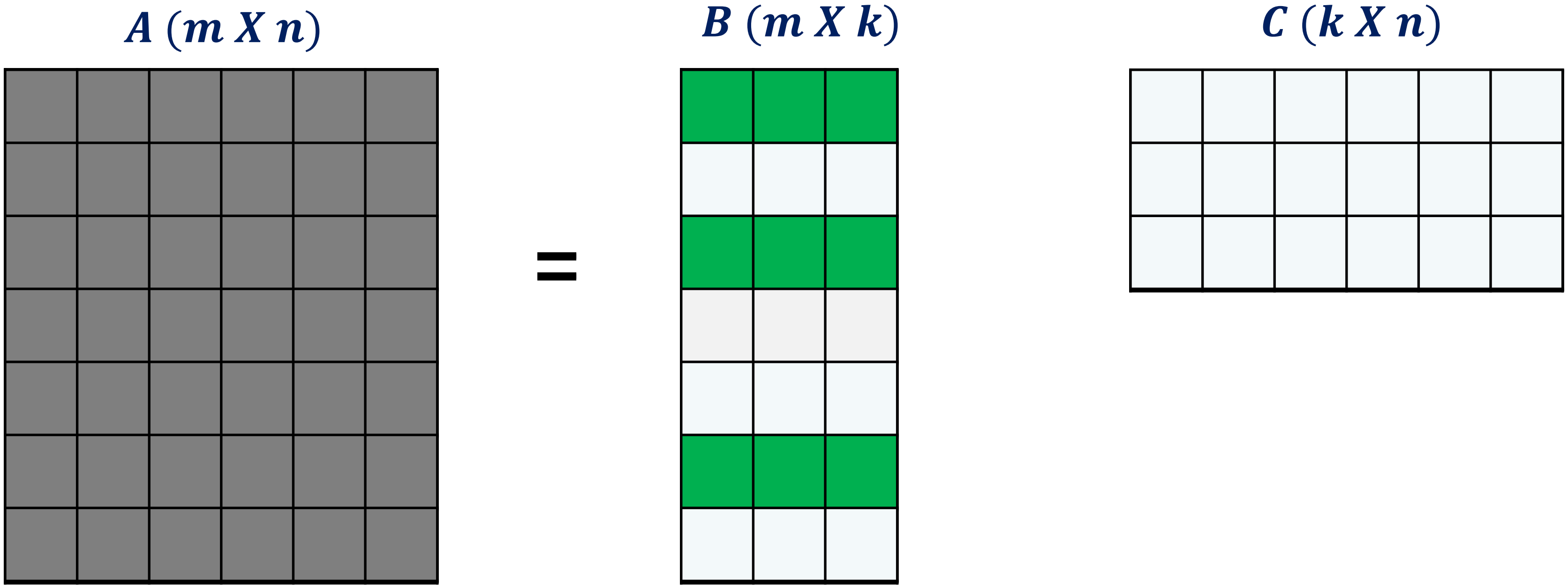
Does there exist binary matrices B and C such that



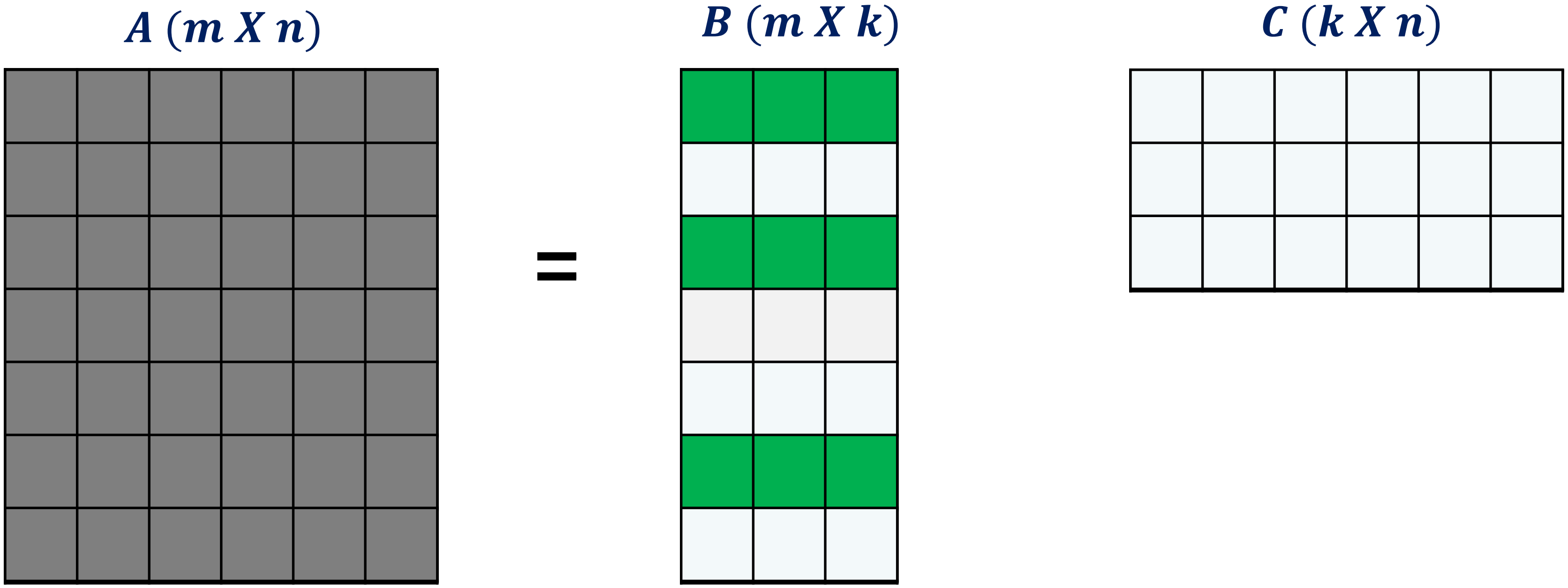
where, product is over standard arithmetic



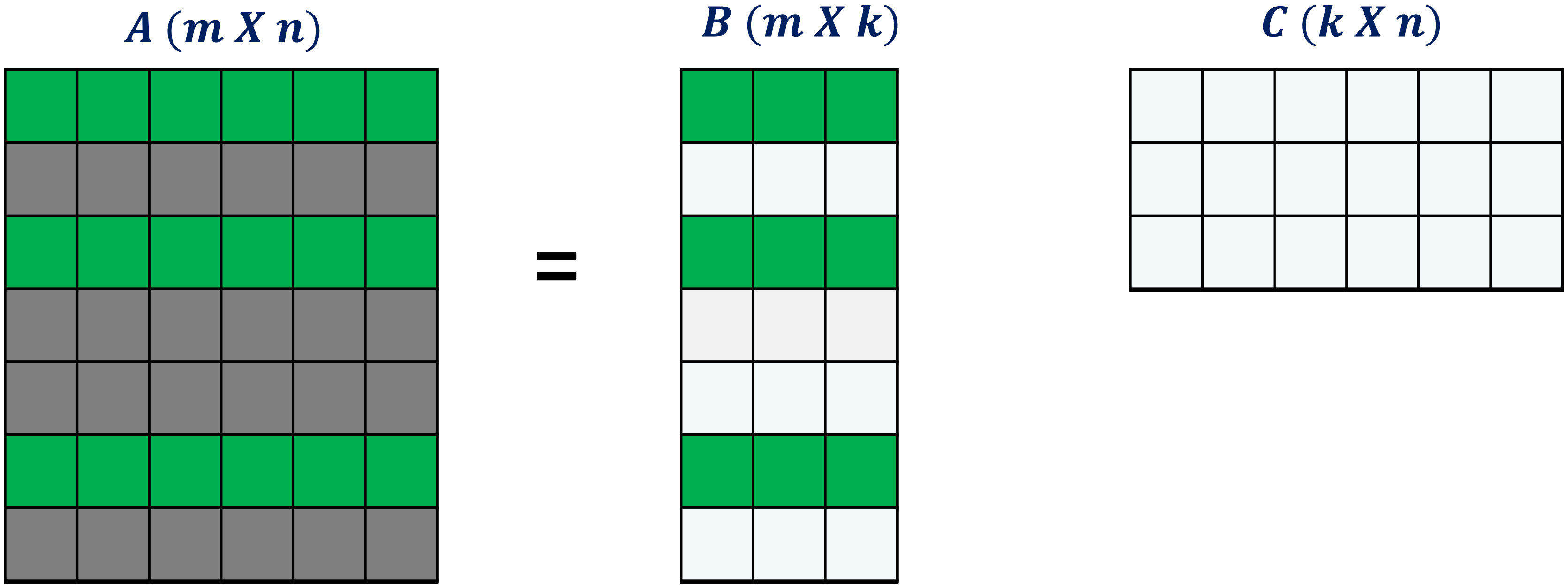
$$m, n \leq 2^k$$



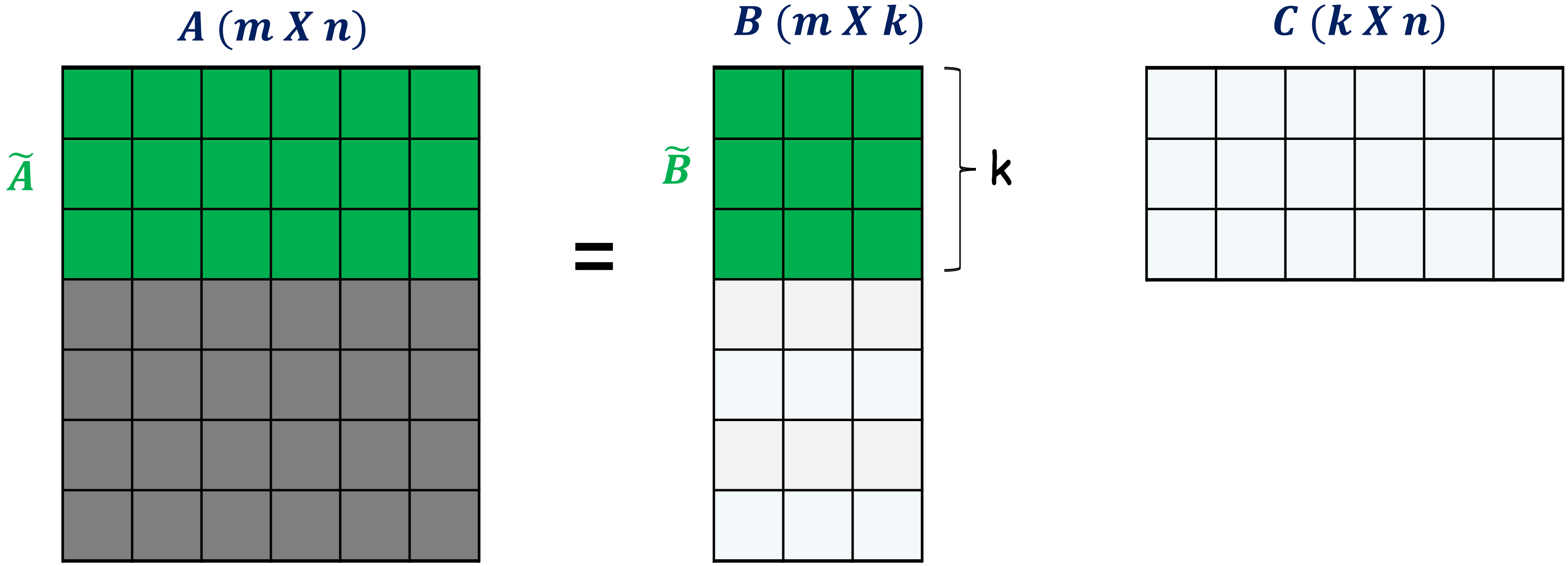
1. Guess indices of k generating rows of B



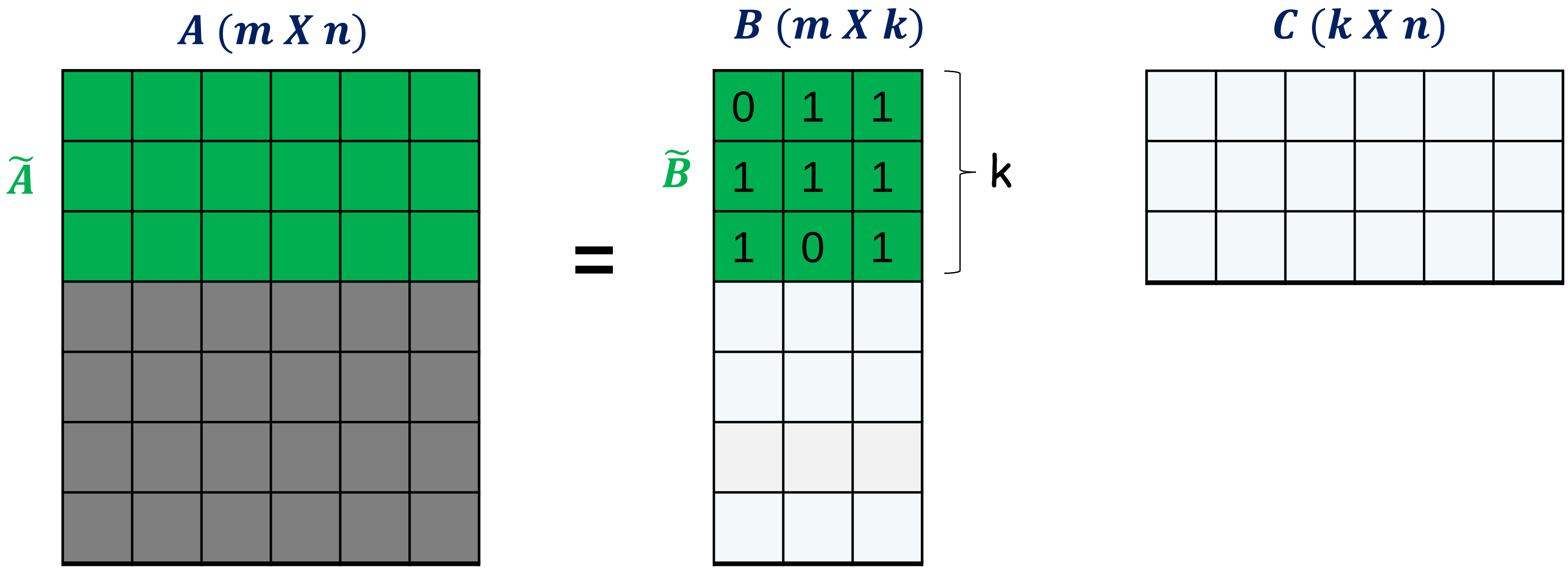
1. Guess indices of k generating rows of B $\binom{m}{k} \leq 2^{k^2}$



1. Guess indices of k generating rows of B $\binom{m}{k} \leq 2^{k^2}$

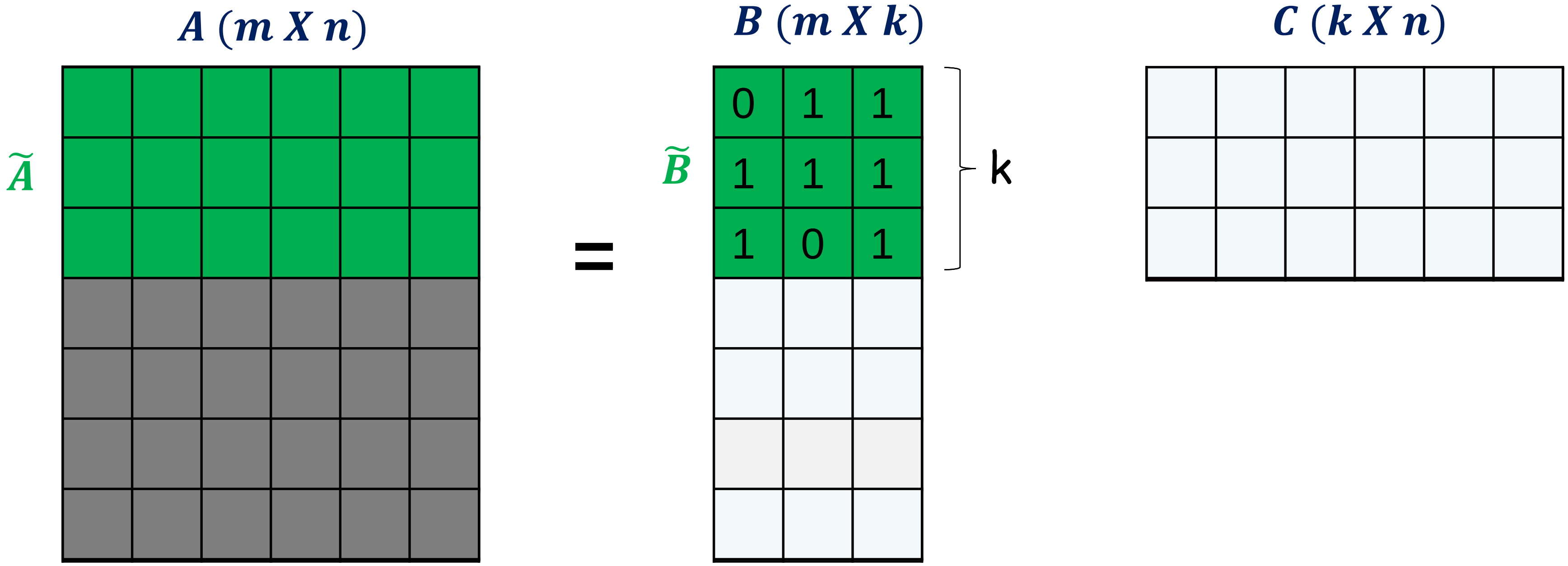


1. Guess indices of k generating rows of B $\binom{m}{k} \leq 2^{k^2}$



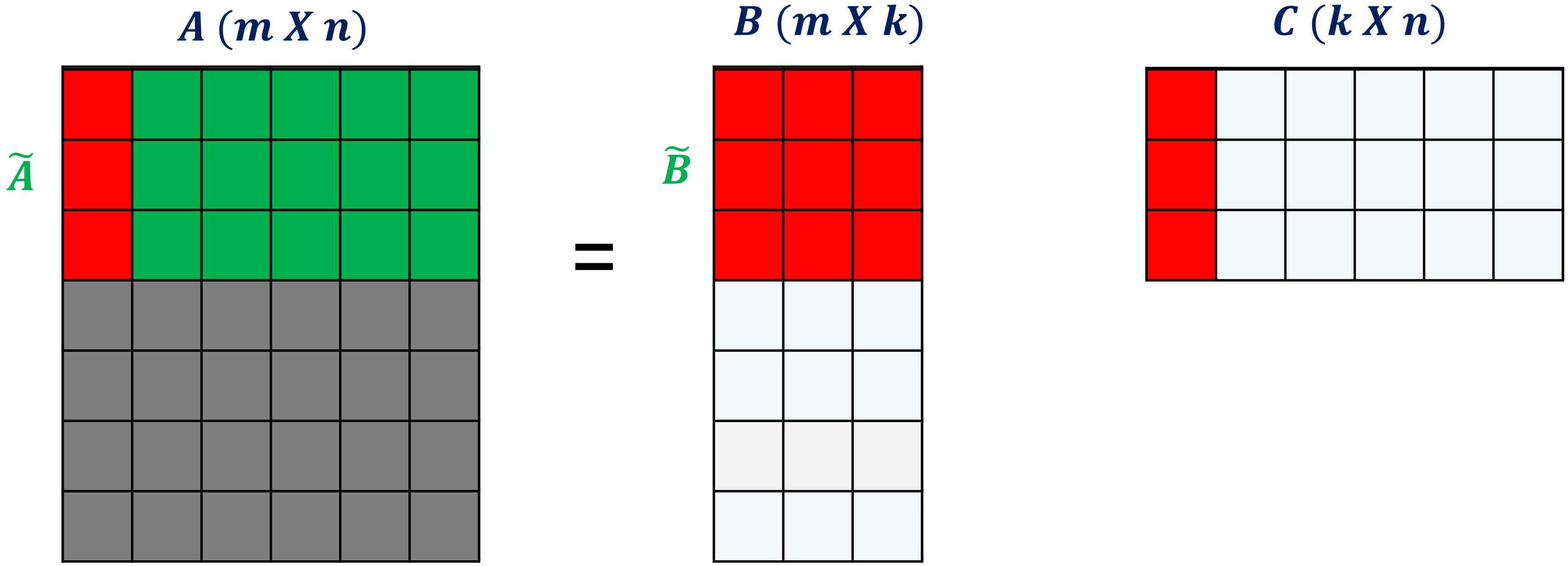
1. Guess indices of k generating rows of B $\binom{m}{k} \leq 2^{k^2}$

2. Guess values of $\tilde{B}$



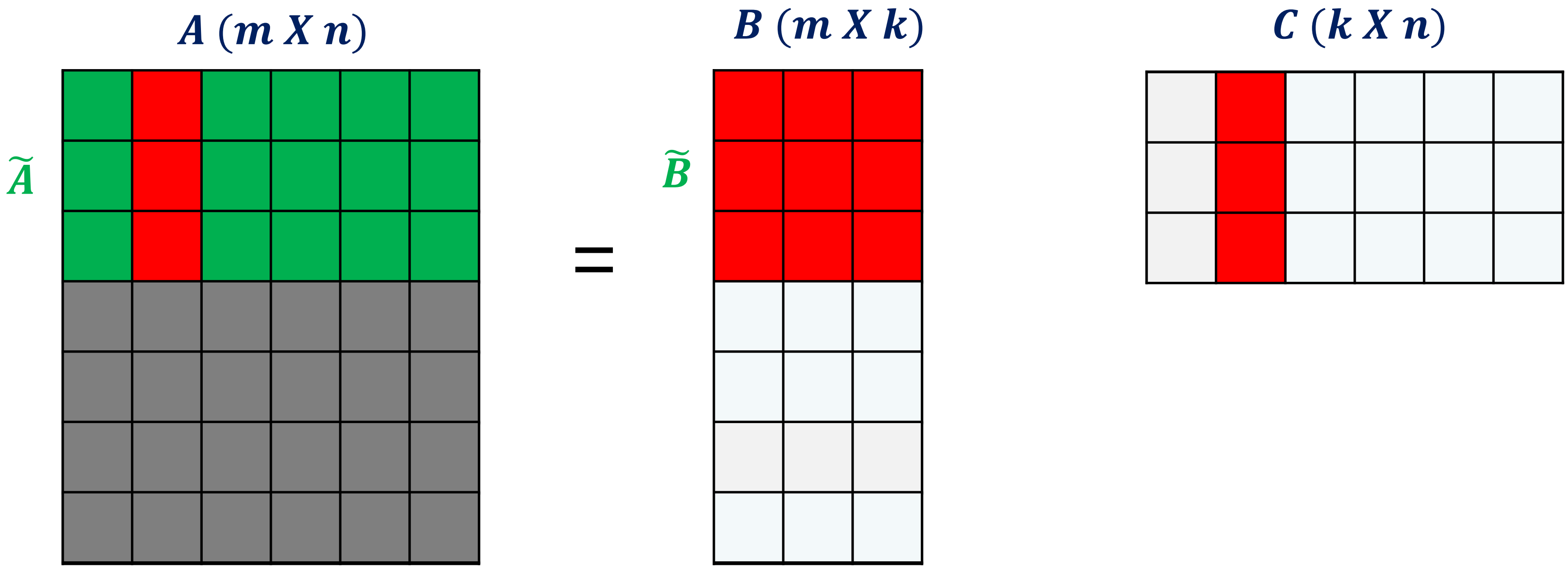
1. Guess indices of k generating rows of B $\binom{m}{k} \leq 2^{k^2}$

2. Guess values of $\tilde{B}$ 2^{k^2}



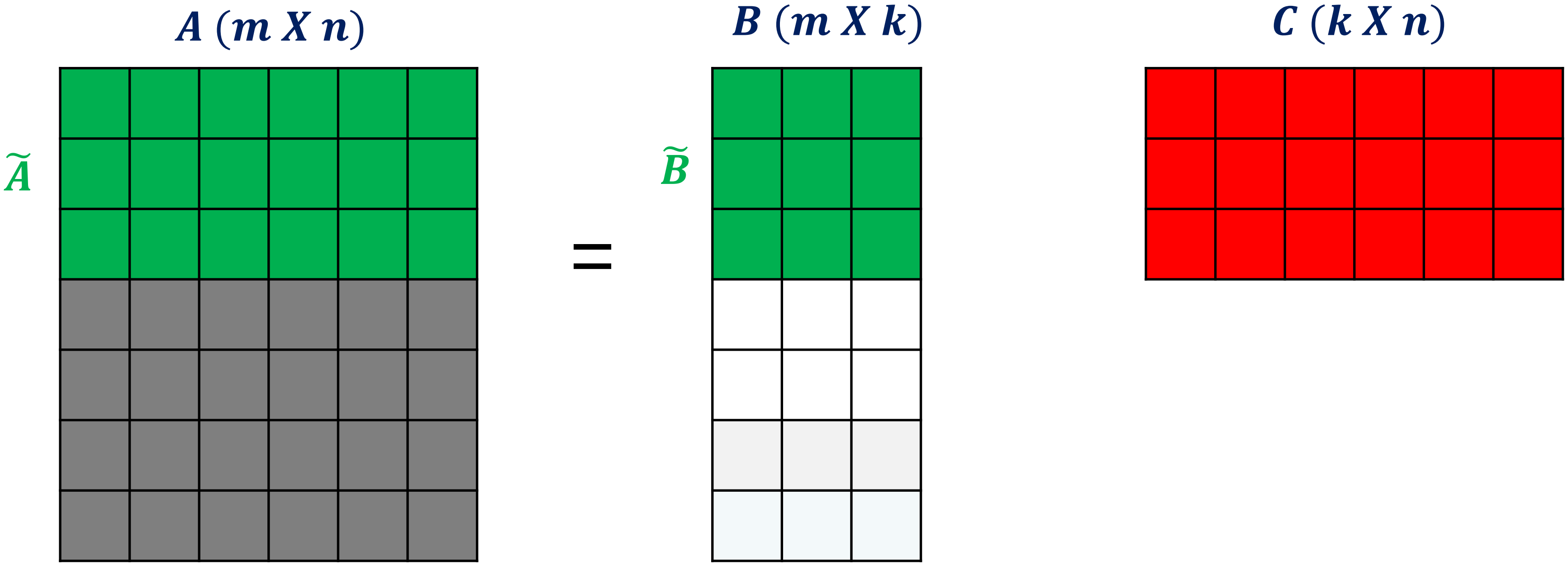
1. Guess indices of $\boldsymbol{k}$ generating rows of $\boldsymbol{B}$ $\quad \binom{m}{k} \leq 2^{k^2}$
2. Guess values of $\widetilde{B}$ $\quad 2^{k^2}$
3. For each $1 \leq i \leq n$,

   find 0-1 column $C_{:\,i}$ s.t $\widetilde{A}_{:\,i} = \widetilde{B}C_{:\,i}$



1. Guess indices of k generating rows of B $\binom{m}{k} \leq 2^{k^2}$
2. Guess values of $\widetilde{B}$ 2^{k^2}
3. For each $1 \leq i \leq n$,

   find 0-1 column $C_{:\,i}$ s.t $\widetilde{A}_{:\,i} = \widetilde{B}C_{:\,i}$

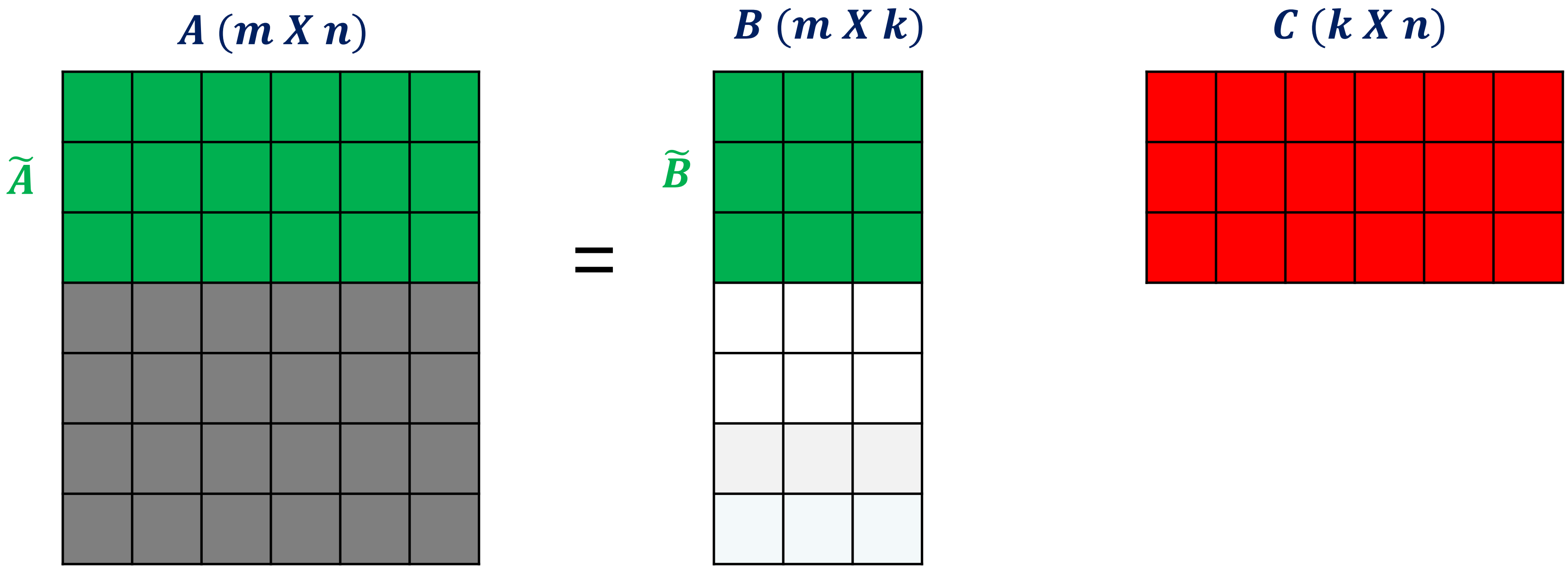


1. Guess indices of k generating rows of B $\binom{m}{k} \leq 2^{k^2}$

2. Guess values of $\tilde{B}$ 2^{k^2}

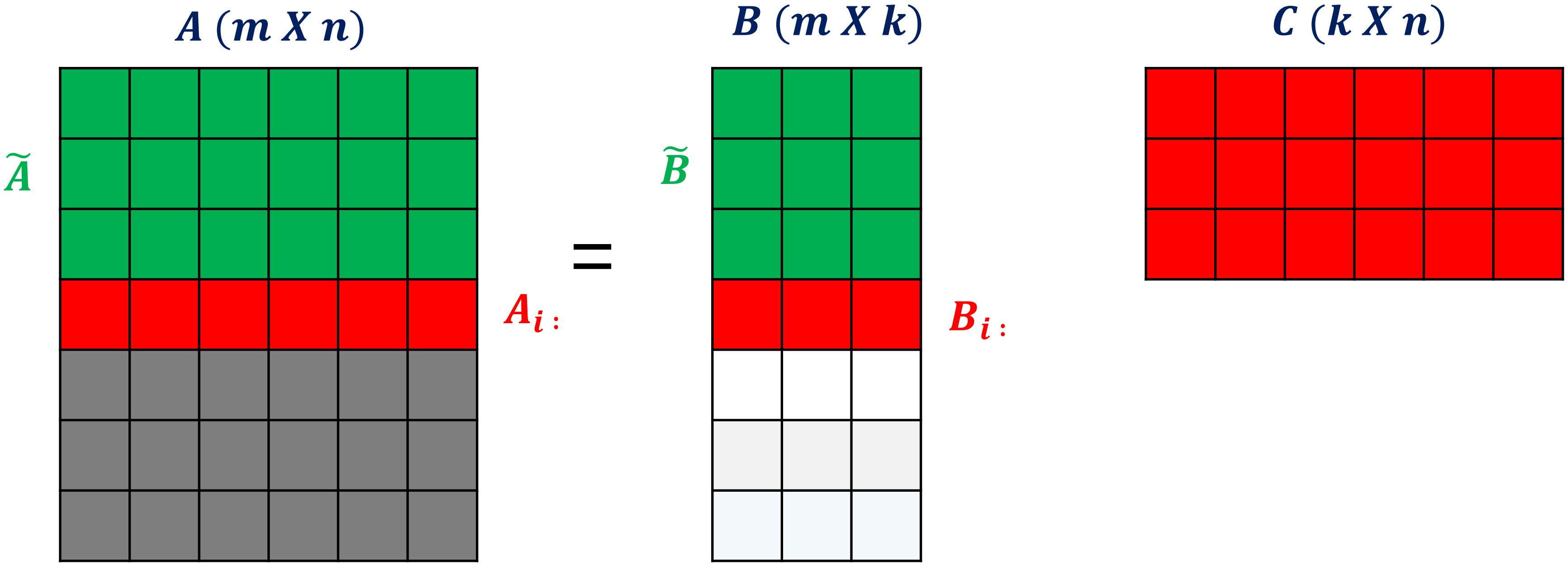
3. For each $1 \leq i \leq n$,

find 0-1 column $C_{:\,i}$ s.t $\tilde{A}_{:\,i} = \tilde{B}C_{:\,i}$



1. Guess indices of $\boldsymbol{k}$ generating rows of $\boldsymbol{B}$ $\binom{m}{k} \leq 2^{k^2}$
2. Guess values of $\widetilde{\boldsymbol{B}}$ 2^{k^2}
3. For each $1 \leq i \leq n$,

   find 0-1 column $\boldsymbol{C}_{:\,\boldsymbol{i}}$ s.t $\widetilde{\boldsymbol{A}}_{:\,\boldsymbol{i}} = \widetilde{\boldsymbol{B}}\boldsymbol{C}_{:\,\boldsymbol{i}}$ $n\,2^k$



1. Guess indices of k generating rows of B $\binom{m}{k} \leq 2^{k^2}$

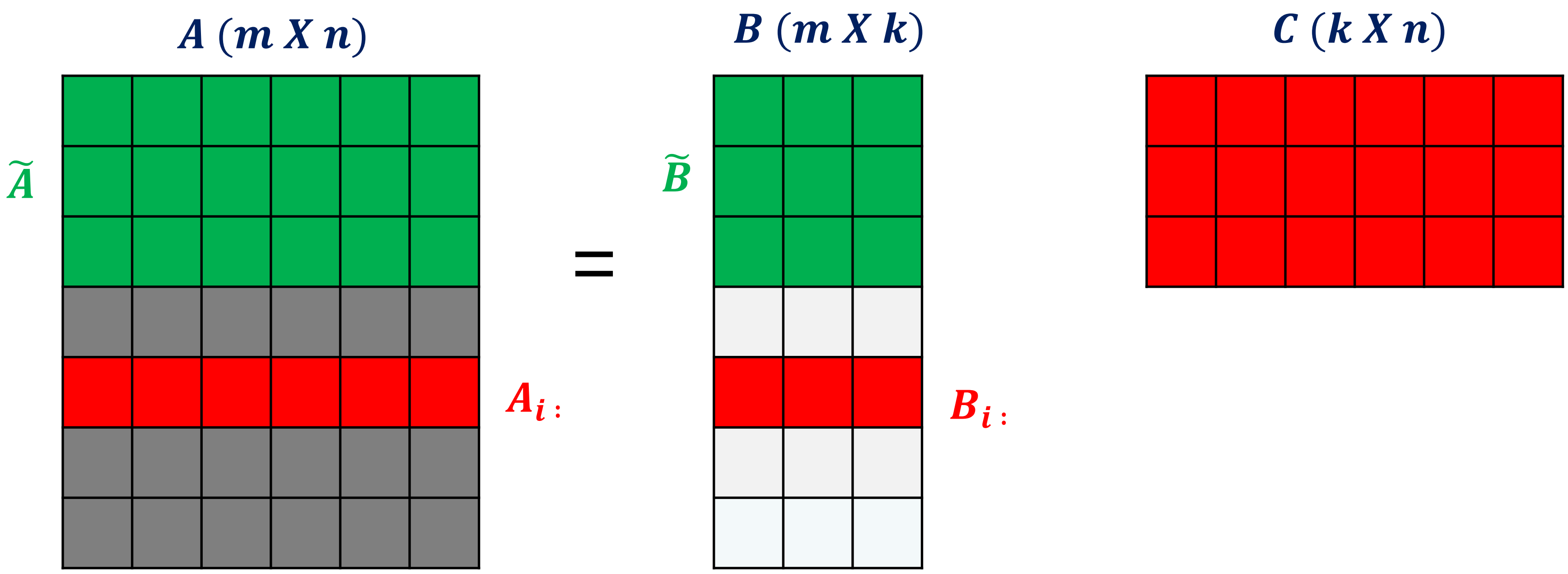
2. Guess values of $\widetilde{B}$ 2^{k^2}

3. For each $1 \leq i \leq n$,

   find 0-1 column $C_{:\,i}$ s.t $\widetilde{A}_{:\,i} = \widetilde{B} C_{:\,i}$ $n\, 2^k$

4. For each $k < i \leq m$,

   find 0-1 row $B_{i\,:}$ s.t $A_{i\,:} = B_{i\,:} C$



1. Guess indices of k generating rows of B $\binom{m}{k} \le 2^{k^2}$

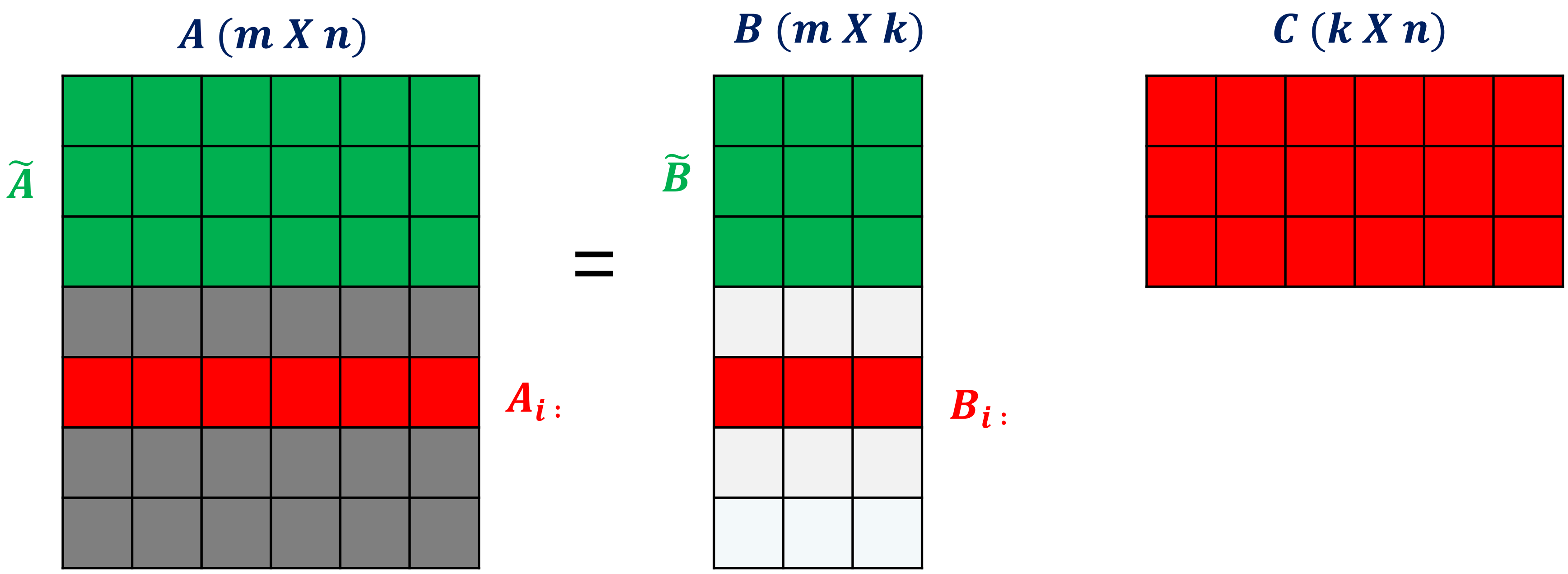
2. Guess values of $\tilde{B}$ 2^{k^2}

3. For each $1 \le i \le n$,

   find 0-1 column $C_{:\,i}$ s.t $\tilde{A}_{:\,i} = \tilde{B} C_{:\,i}$ $n\, 2^k$

4. For each $k < i \le m$,

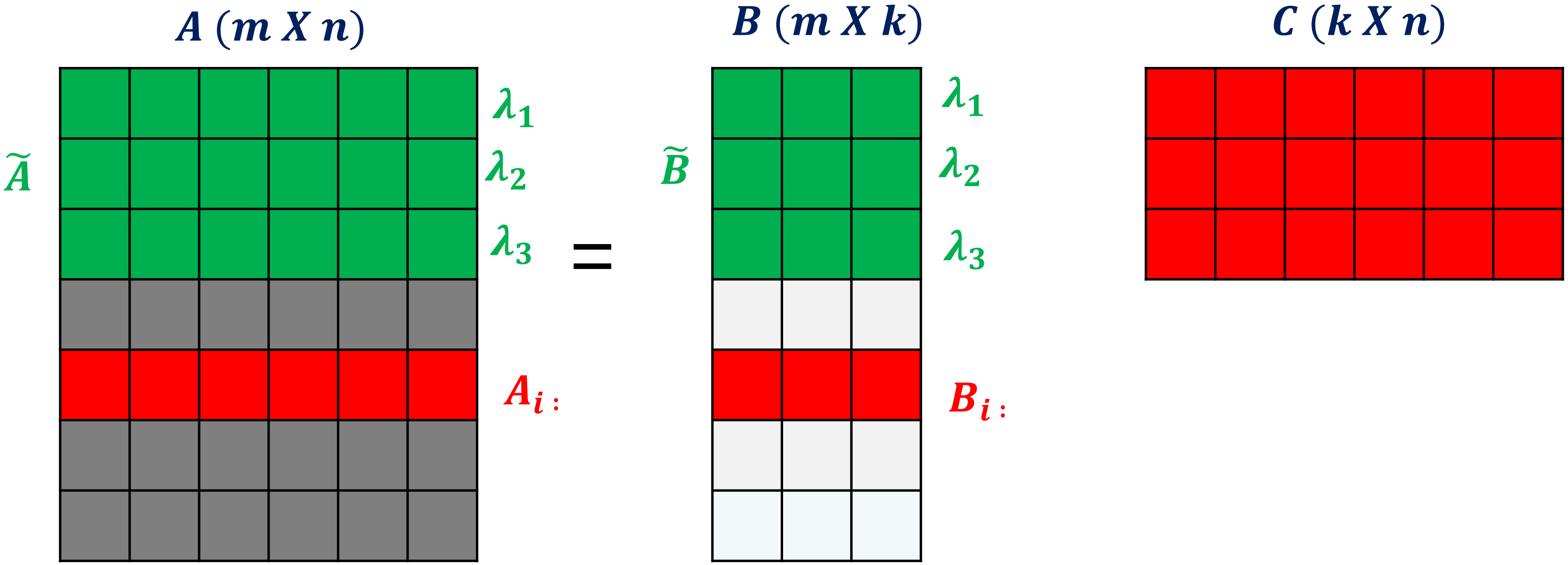
   find 0-1 row $B_{i\,:}$ s.t $A_{i\,:} = B_{i\,:} C$



1. Guess indices of $\boldsymbol{k}$ generating rows of $\boldsymbol{B}$ $\quad \binom{m}{k} \leq 2^{k^2}$
2. Guess values of $\widetilde{\boldsymbol{B}}$ $\quad 2^{k^2}$
3. For each $1 \leq i \leq n$,

   find 0-1 column $\boldsymbol{C}_{:\,i}$ s.t $\widetilde{\boldsymbol{A}}_{:\,i} = \widetilde{\boldsymbol{B}}\boldsymbol{C}_{:\,i}$ $\quad n\, 2^k$
4. For each $k < i \leq m$,

   find 0-1 row $\boldsymbol{B}_{i\,:}$ s.t $\boldsymbol{A}_{i\,:} = \boldsymbol{B}_{i\,:}\boldsymbol{C}$ $\quad m\, 2^k$



1. Gu

2. Gu

3. Fo

4. Fo

Why should such $B_{i\,:}$ exist ?

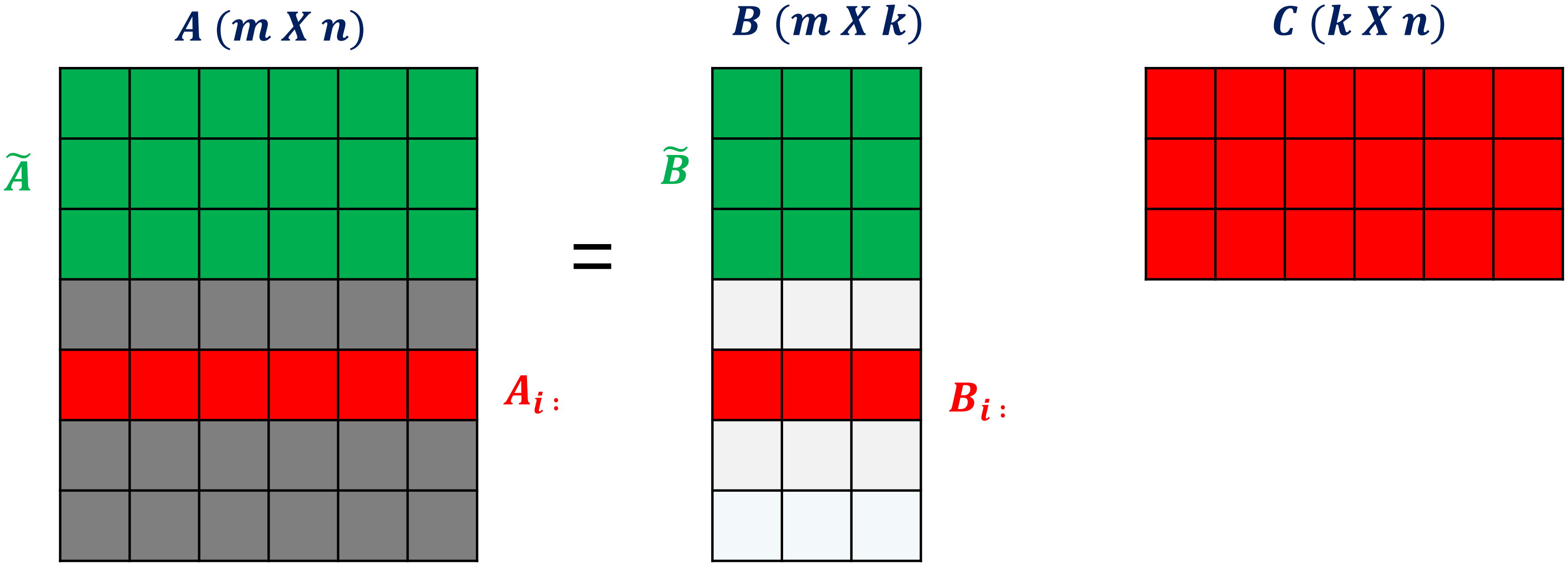
- Let $B^*,\ C^*$ be a solution
- <u>Claim.</u> $B^*_{i\,:}$ is the required $B_{i\,:}$
- $B^*_{i\,:}C = \left(\sum_{1\le j\le k} \lambda_j B_{j\,:}\right) C$
- $= \sum_{1\le j\le k} \lambda_j A_{j\,:}$
- $= A_{i\,:}$ (using $A = B^* C^*$)

$\binom{m}{k} \le 2^{k^2}$

2^{k^2}

$n\ 2^k$

$m\ 2^k$



1. Guess indices of k generating rows of B $\qquad \binom{m}{k} \le 2^{k^2}$

2. Guess values of $\tilde{B}$ $\qquad 2^{k^2}$

3. For each $1 \le i \le n$,

   find 0-1 column $C_{:i}$ s.t $\tilde{A}_{:i} = \tilde{B} C_{:i}$ $\qquad n\, 2^k$

4. For each $k < i \le m$,

   find 0-1 row $B_{i:}$ s.t $A_{i:} = B_{i:} C$ $\qquad m\, 2^k$

Generalizations of the algorithm

- Weighted biclique partition

  input matrix is integer instead of binary
- Arithmetic over any Field $\boldsymbol{F}$
- If the entries of B and C are allowed to be

  from some $\boldsymbol{S} \subseteq \boldsymbol{F}$, the algorithm will work

  with running time $\boldsymbol{O}^{*}\left(|\boldsymbol{S}|^{3k^{2}}\right)$
- Edge Clique Partition

Open Problems

1. Polynomial kernel for Biclique partition?
2. Parameterized approximation algorithms for Biclique Cover
3. Approximate kernels for Bilcique cover
4. Is the $2^{O(k^2)}$ running time optimal for Bipartite biclique partition?
5. FPT algorithm for weighted edge clique partition?

Thank You for Listening !!! Questions??