Bipartite Graphs and Matchings

Ema Skottova, Karl Robert, Benjamin Unger

Primes-Switzerland, Mentor: Kaloyan Slavov

23.6.18

König's Theorem

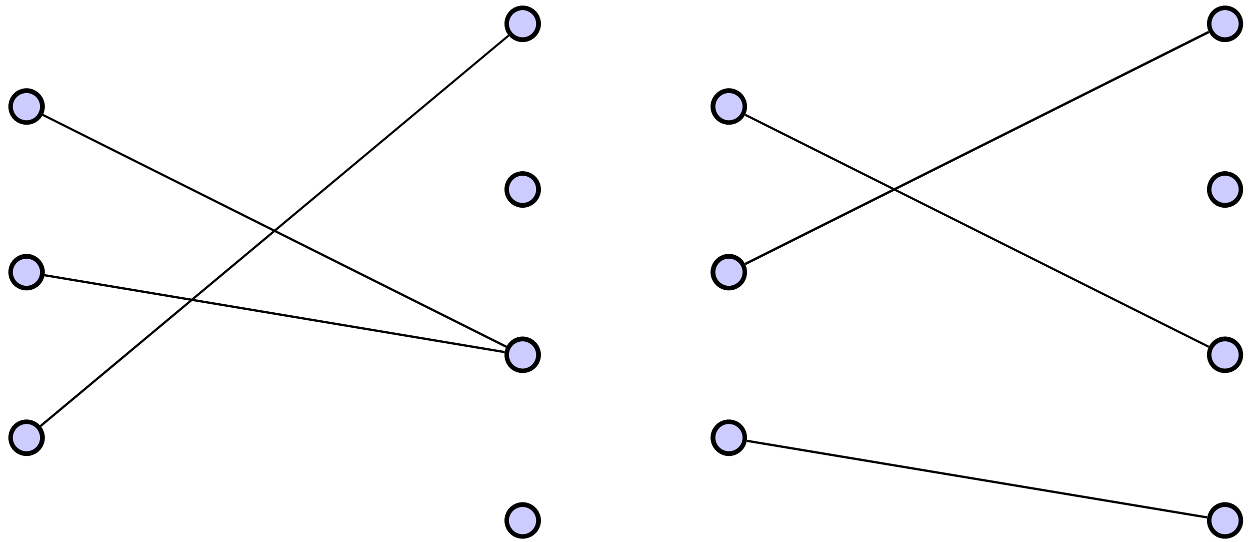
Benjamin Unger

KEN 5th year Highschool

23.6.18

Introduction

- Bipartite Graph: Graph whose vertices can be partitioned into two different independent sets U and V such that no edges are between any two vertices in either set.
- Perfect matching of U into V: Set of edges without common vertices. Where every vertex in U is connected to an edge which goes to a distinct vertex in V



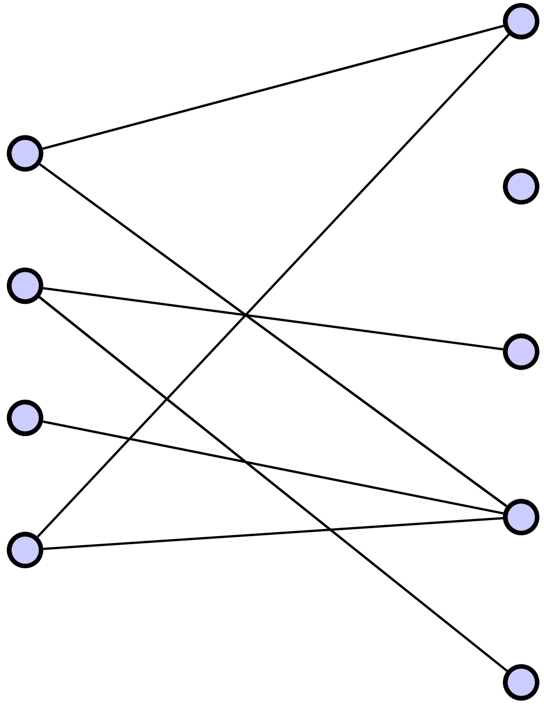
Problem

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

a = minimum # of lines that cover all 1's
b = maximum # of independent 1's
Prove that a = b.

Solution

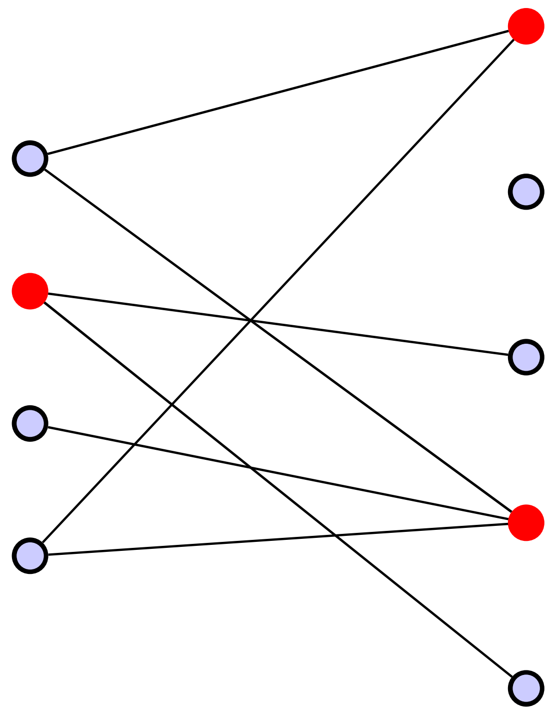
$$\begin{matrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 \end{matrix}$$



To prove this we use a bipartite graph and translate the 1's and 0's to edges and the rows and columns to vertices.

Solution

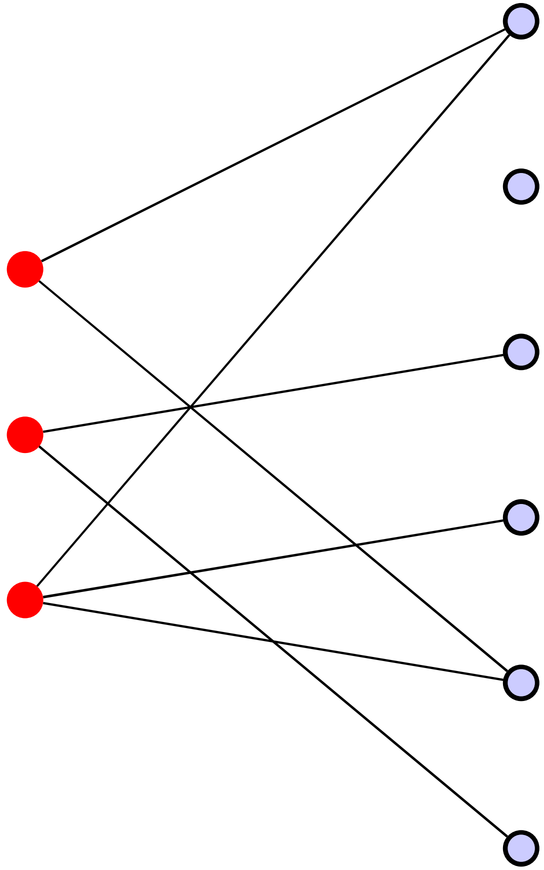
1	0	0	1	0
0	0	1	0	1
0	0	0	1	0
1	0	0	1	0



We color the minimum number of lines/vertices and take all of them to the left side and all others to the right.

Solution

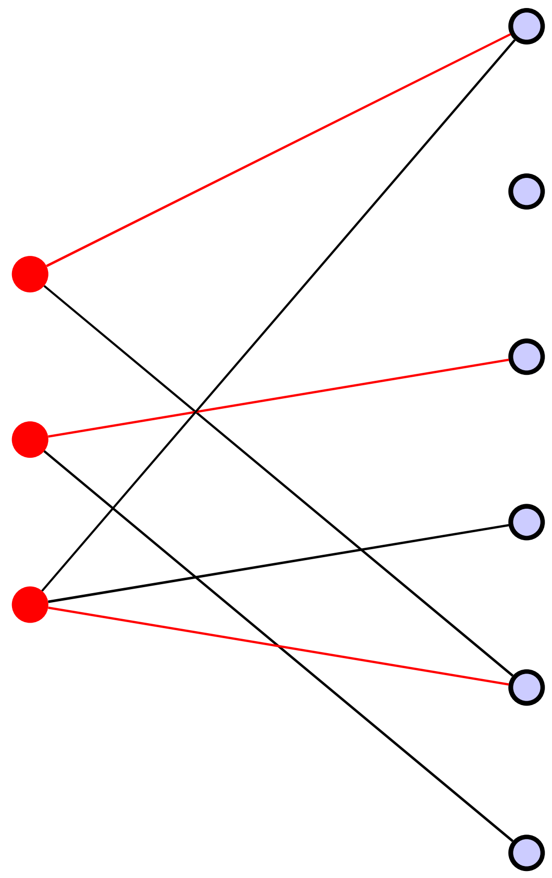
$$\begin{array}{ccccc} 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 \end{array}$$



Now we get a new bipartite graph in witch we have to find a matching from U into V. This will represent a set of independent 1's.

Solution

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



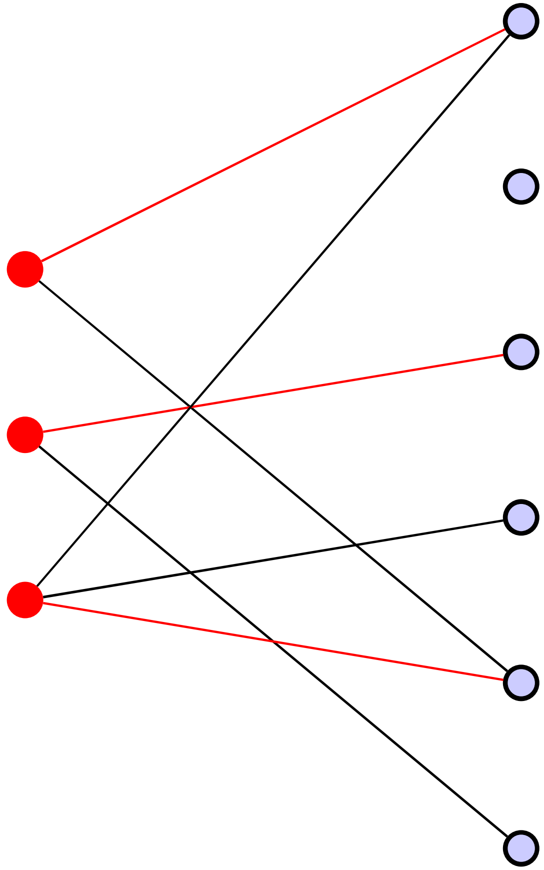
We need to prove that a matching exists. For this we use Phillip Hall's Theorem.

Philip Hall's Theorem

Let G be a bipartite graph with bipartite sets X and Y. For a set W of vertices in X, let $N_G(W)$ denote the neighborhood of W in G, i.e. the set of all vertices in Y adjacent to some element of W. The theorem in this formulation states that there is a matching that entirely covers X if and only if for every subset W of X: $|W| \leq |N_G(W)|$.

Solution

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



Proof by contradiction: Suppose there exists a subset W where $|W| > |N_G(W)|$.Then we could just switch W with $N_G(W)$ and get a smaller set of lines! From this follows that b $=$ a! Q.E.D.

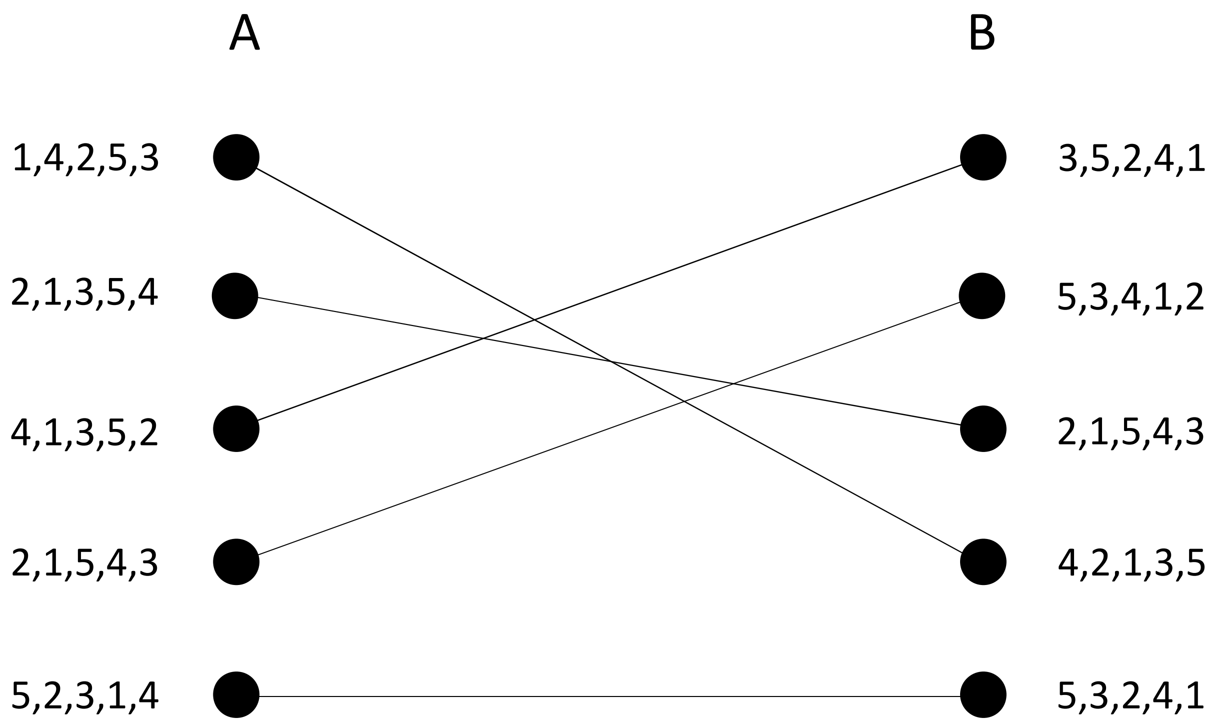
Stable Matchings

Ema Skottova

Gymnasium Kirchenfeld Bern

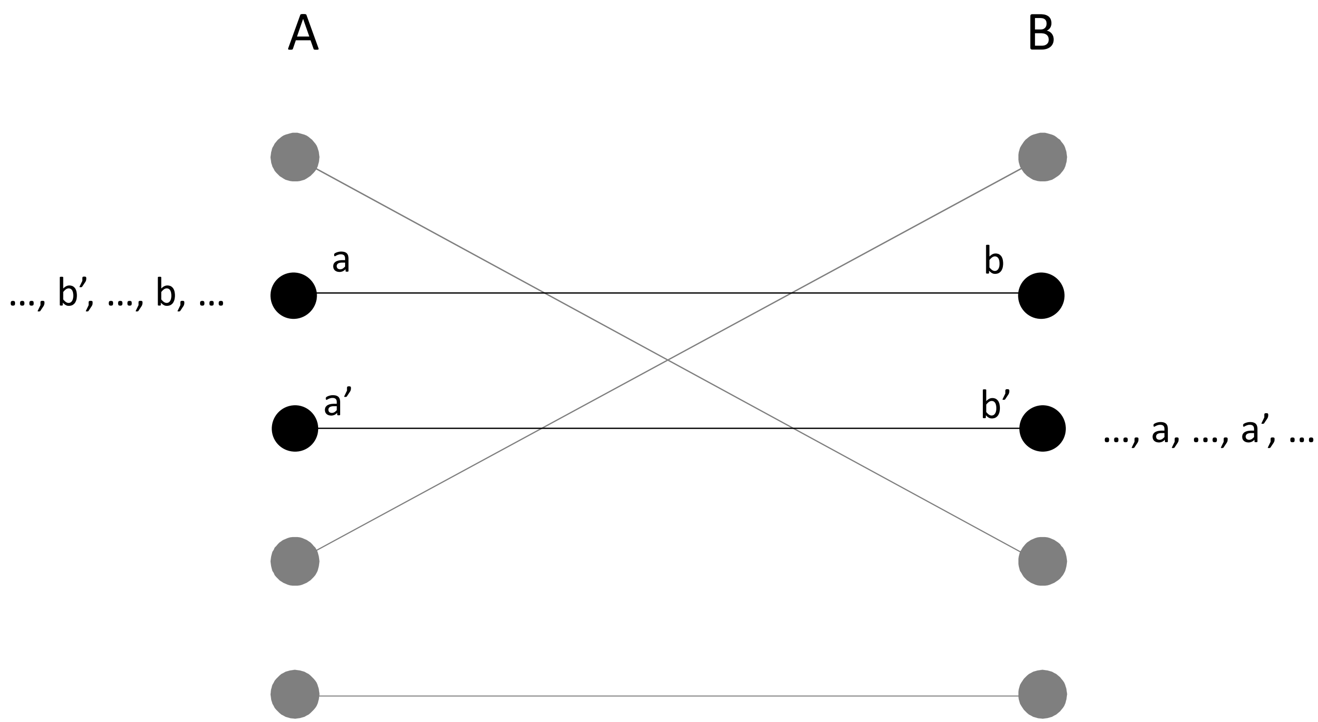
Year 9

Requirements for stable matchings



- Bipartite Graph
- |A| = |B|
- Preferences

Definition of a stable matching



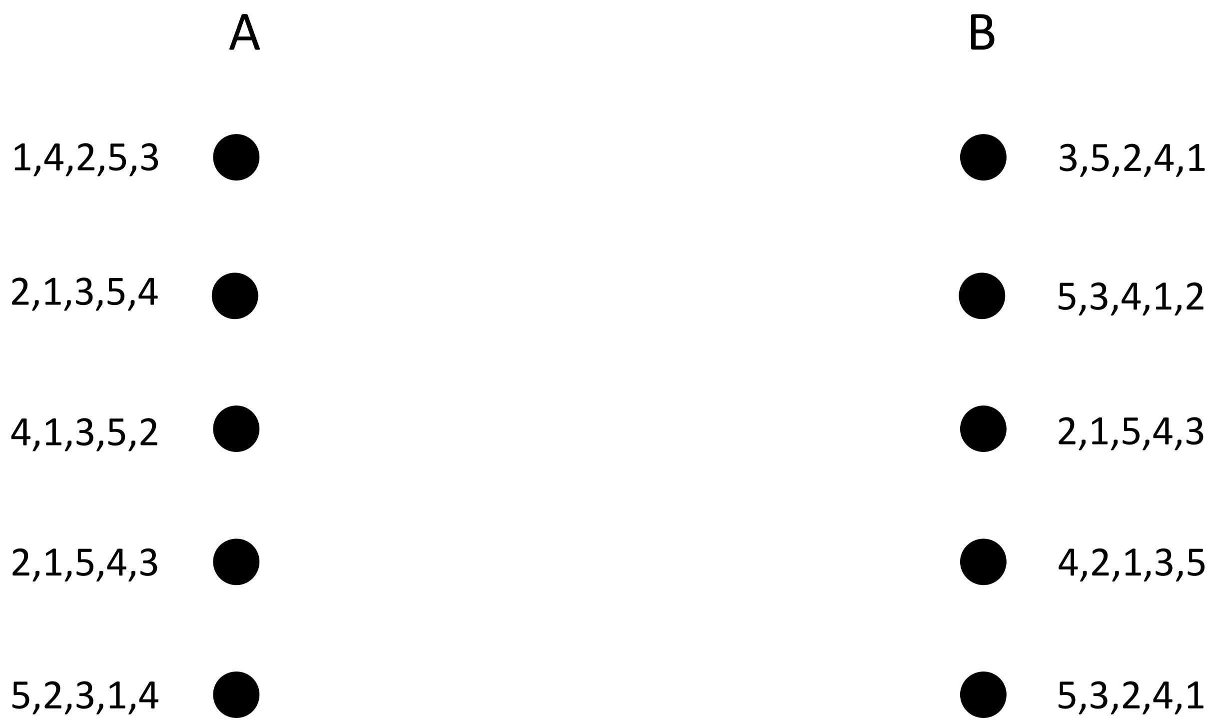
A perfect matching is not stable if ∃ vertices a, b, a' and b' such that:

- There is an edge ab but a prefers b'

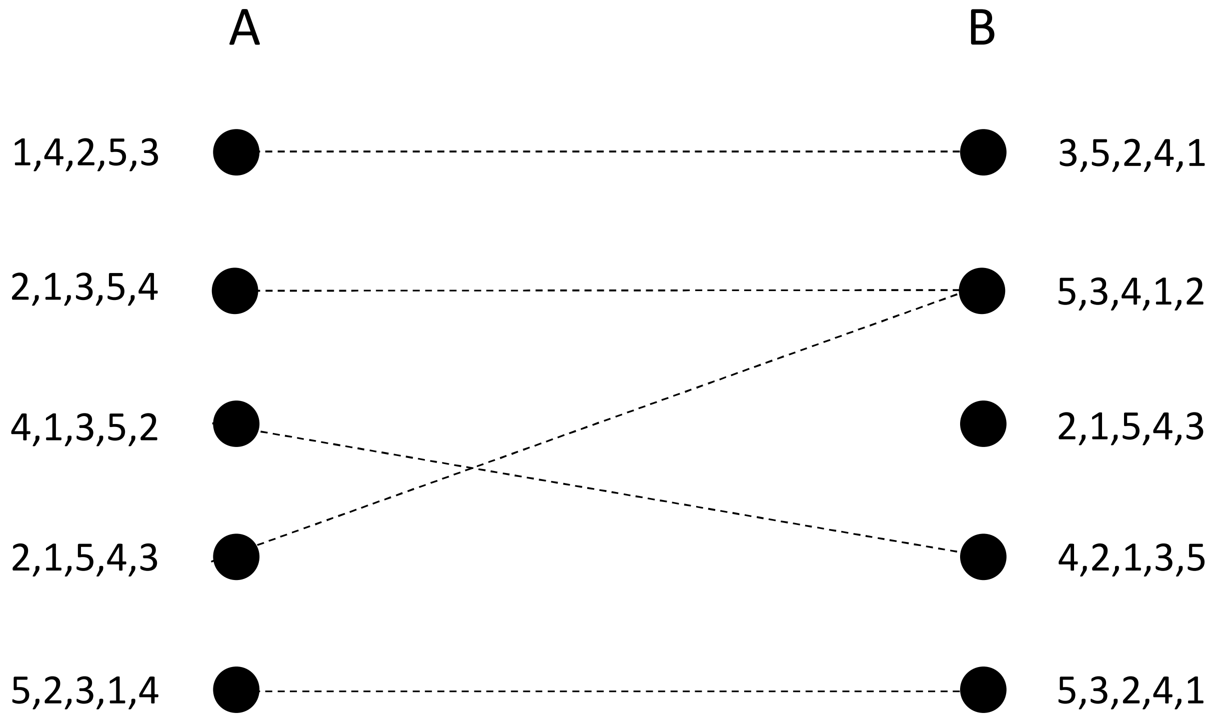
and

- There is an edge a'b' but b' prefers a

Gale-Shapley Algorithm

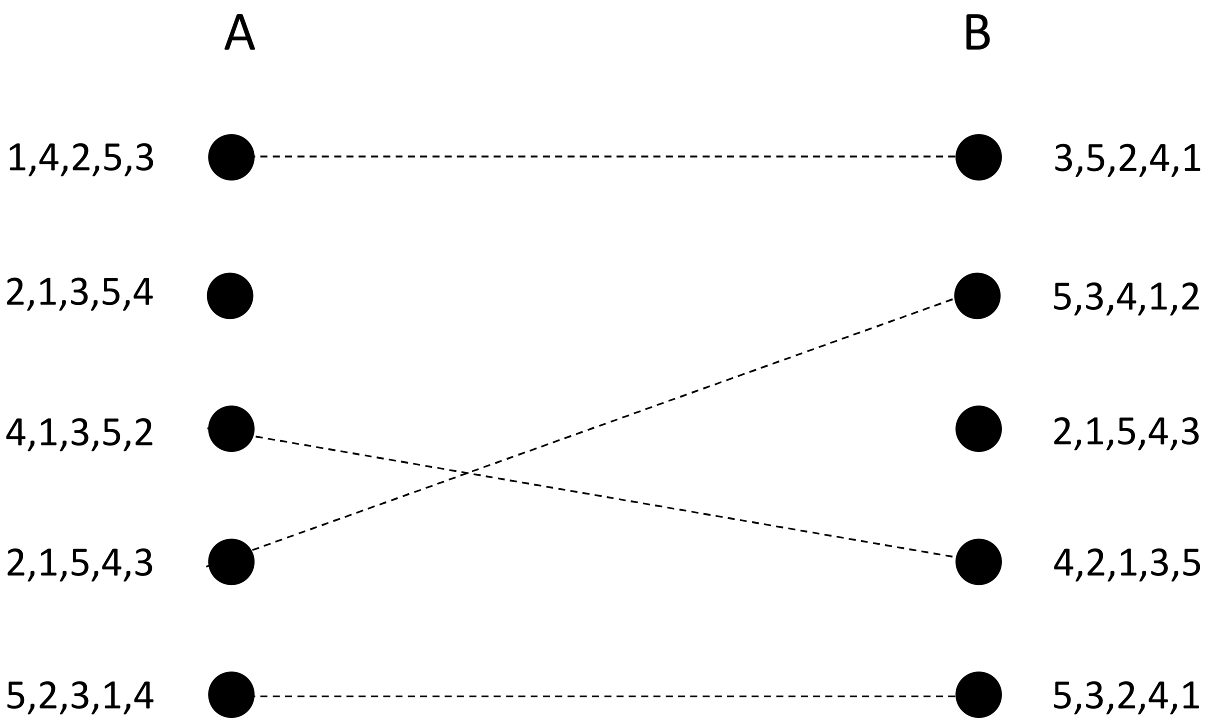


Gale-Shapley Algorithm



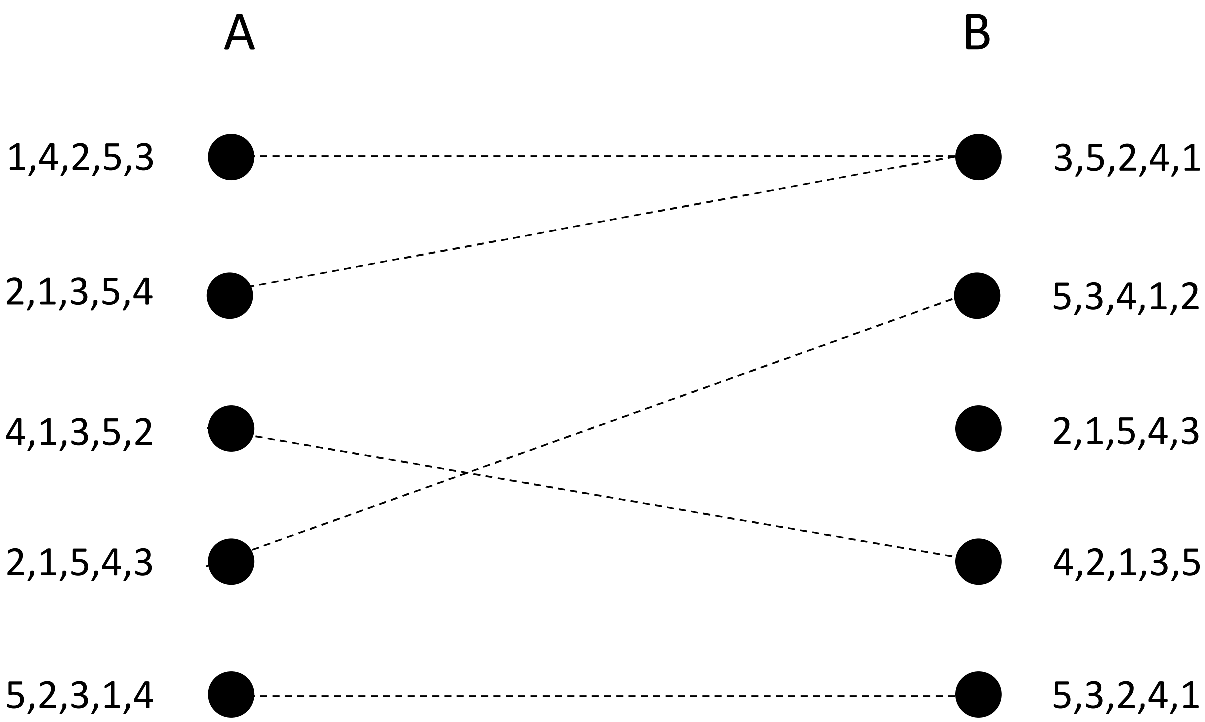
1. Applicants apply for favorite job.

Gale-Shapley Algorithm



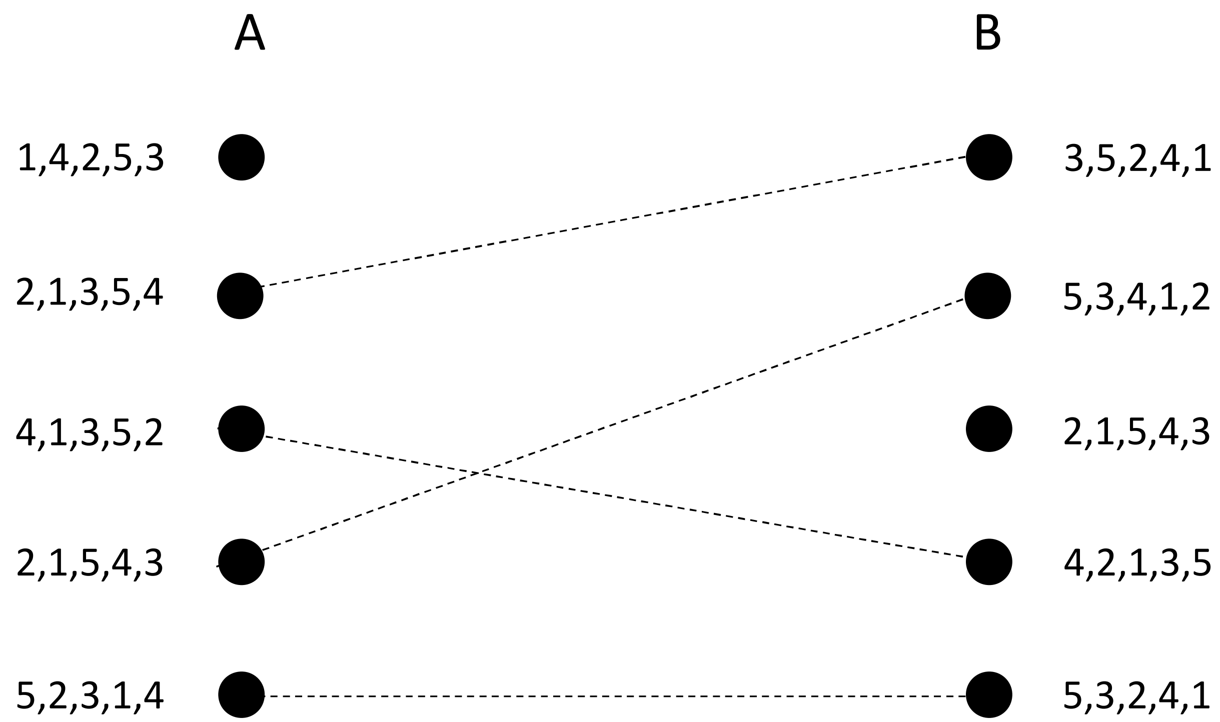
1. Applicants apply for favorite job.
2. Preferred applicants are hired temporarily, rest rejected.

Gale-Shapley Algorithm



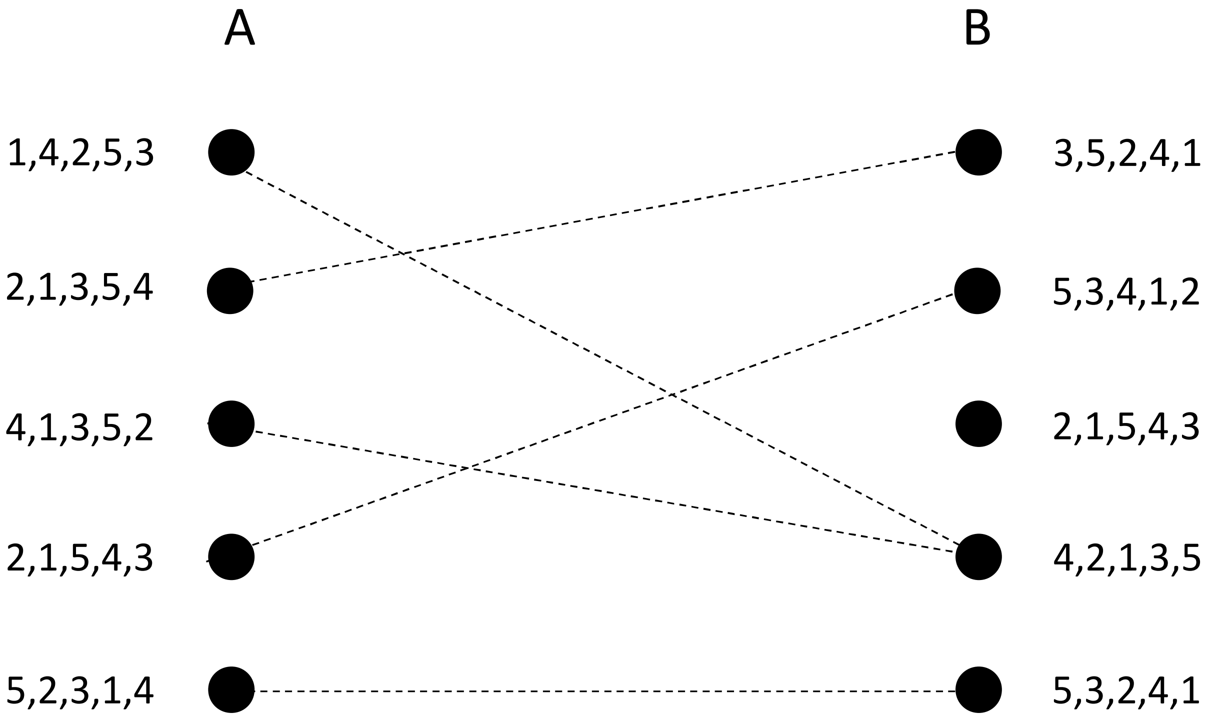
1. Applicants apply for favorite job.
2. Preferred applicants are hired temporarily, rest rejected.
3. Applicants that aren’t hired temporarily apply for next choice.

Gale-Shapley Algorithm



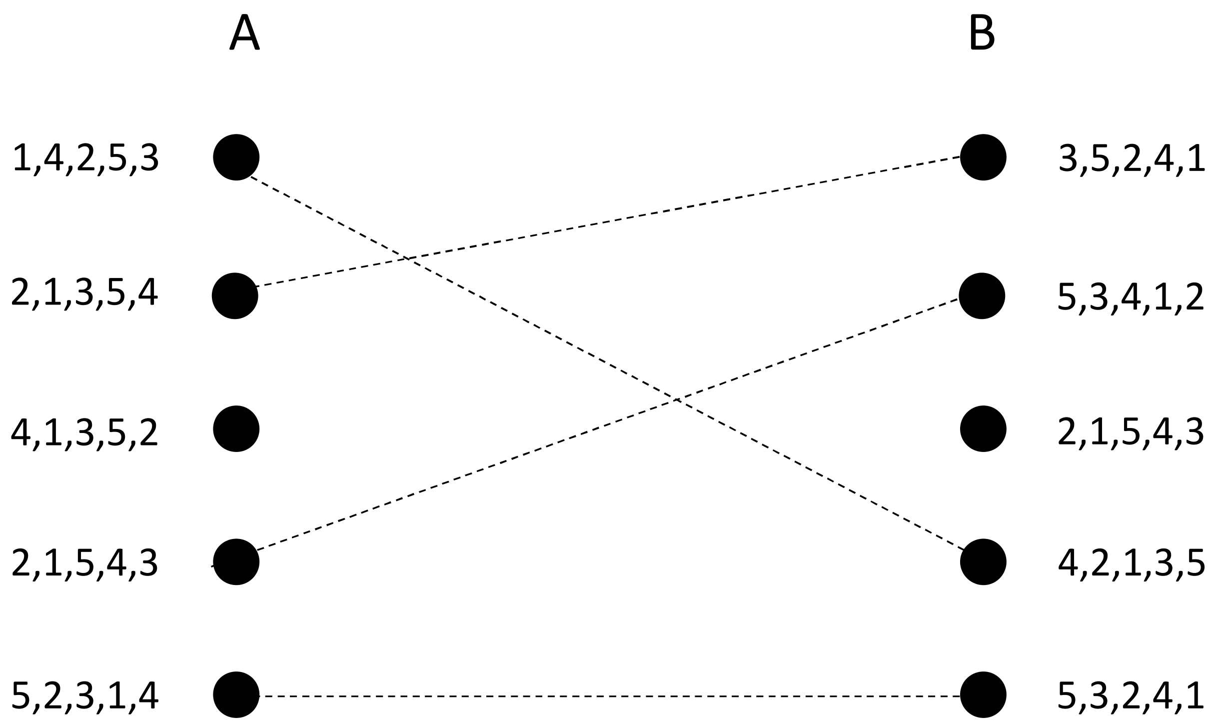
1. Applicants apply for favorite job.
2. Preferred applicants are hired temporarily, rest rejected.
3. Applicants that aren't hired temporarily apply for next choice.
4. Repeat 2 and 3 until everyone is hired.

Gale-Shapley Algorithm



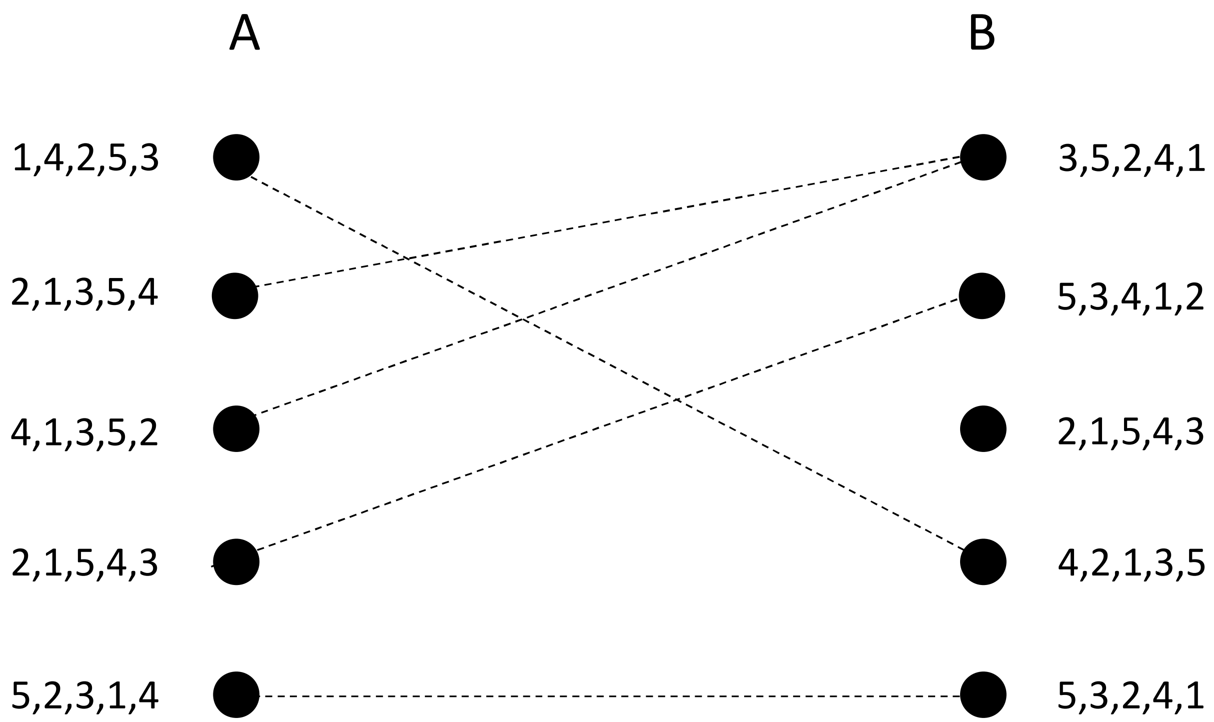
1. Applicants apply for favorite job.
2. Preferred applicants are hired temporarily, rest rejected.
3. Applicants that aren't hired temporarily apply for next choice.
4. Repeat 2 and 3 until everyone is hired.

Gale-Shapley Algorithm



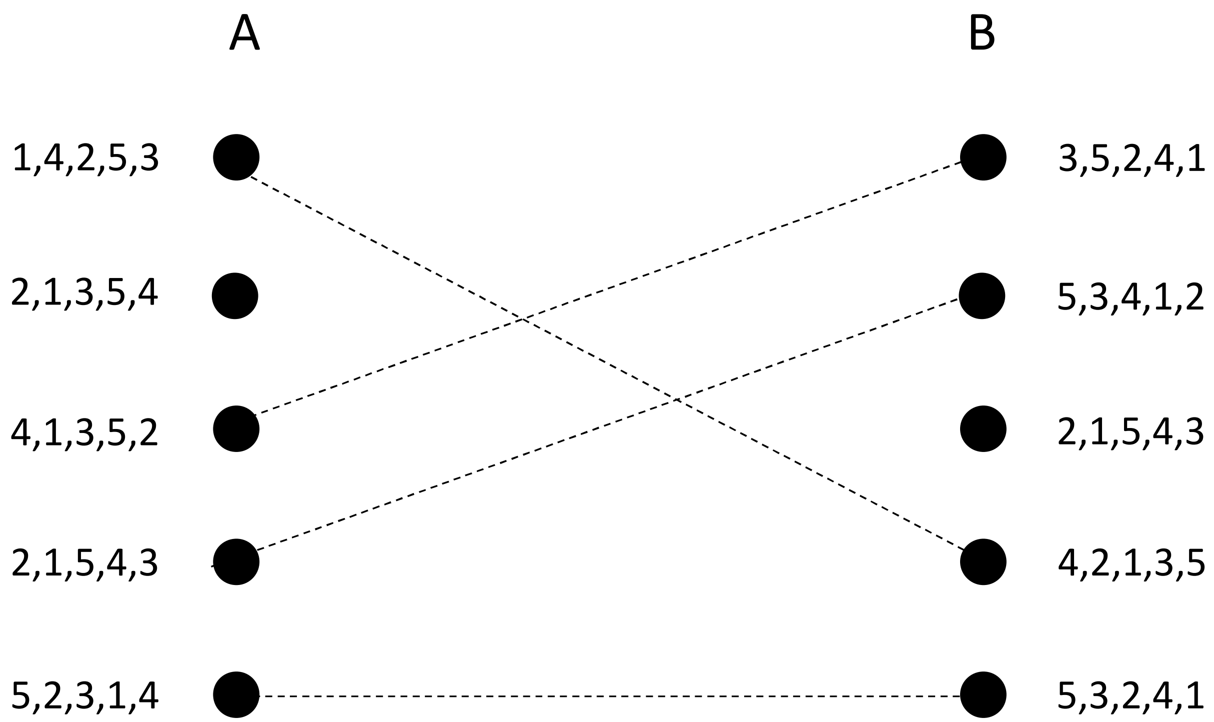
1. Applicants apply for favorite job.
2. Preferred applicants are hired temporarily, rest rejected.
3. Applicants that aren't hired temporarily apply for next choice.
4. Repeat 2 and 3 until everyone is hired.

Gale-Shapley Algorithm



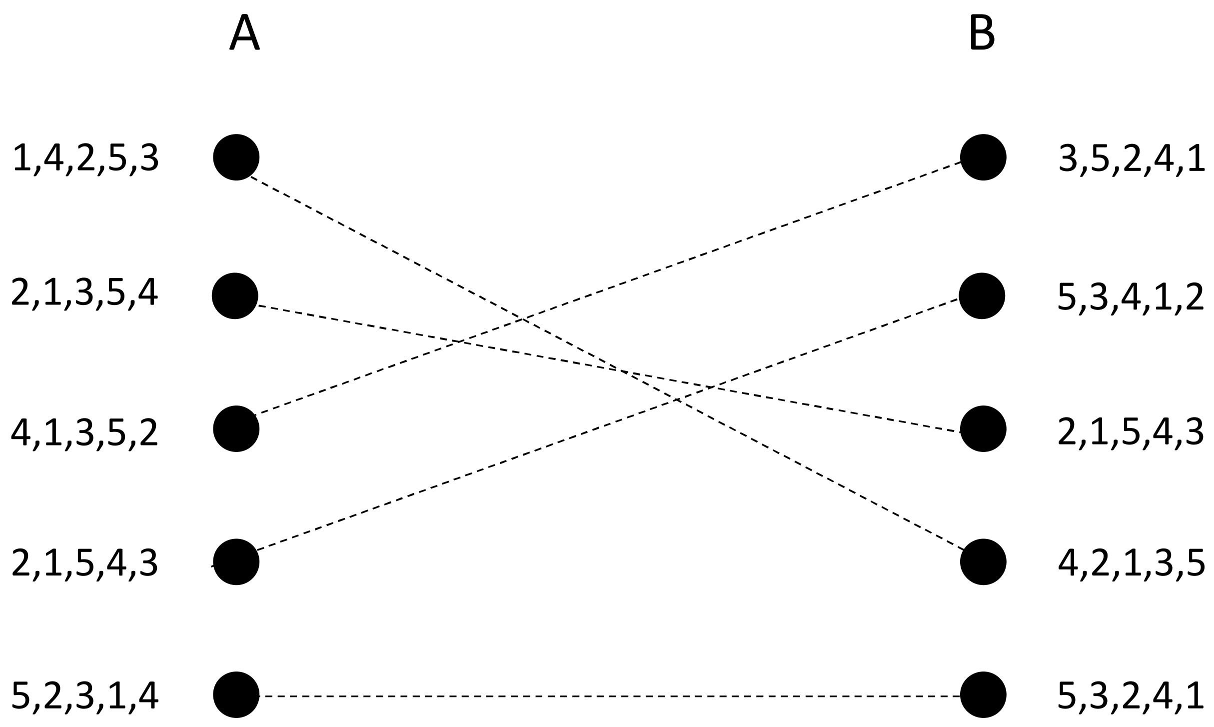
1. Applicants apply for favorite job.
2. Preferred applicants are hired temporarily, rest rejected.
3. Applicants that aren't hired temporarily apply for next choice.
4. Repeat 2 and 3 until everyone is hired.

Gale-Shapley Algorithm



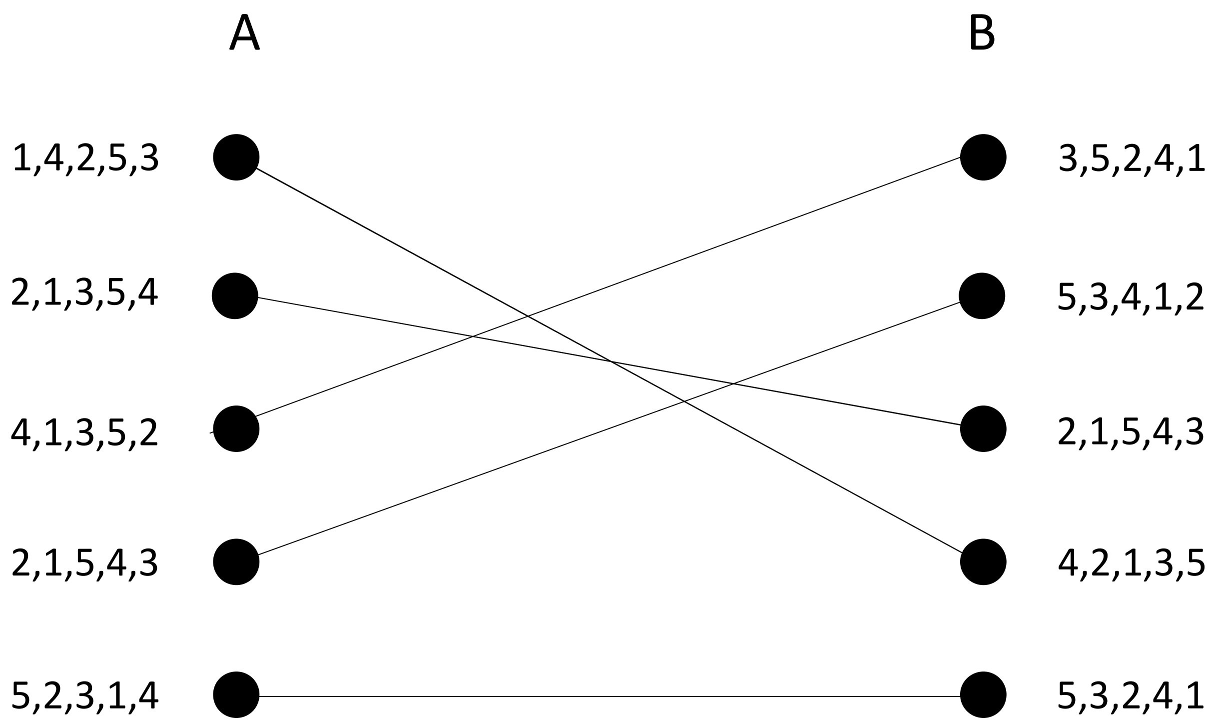
1. Applicants apply for favorite job.
2. Preferred applicants are hired temporarily, rest rejected.
3. Applicants that aren't hired temporarily apply for next choice.
4. Repeat 2 and 3 until everyone is hired.

Gale-Shapley Algorithm



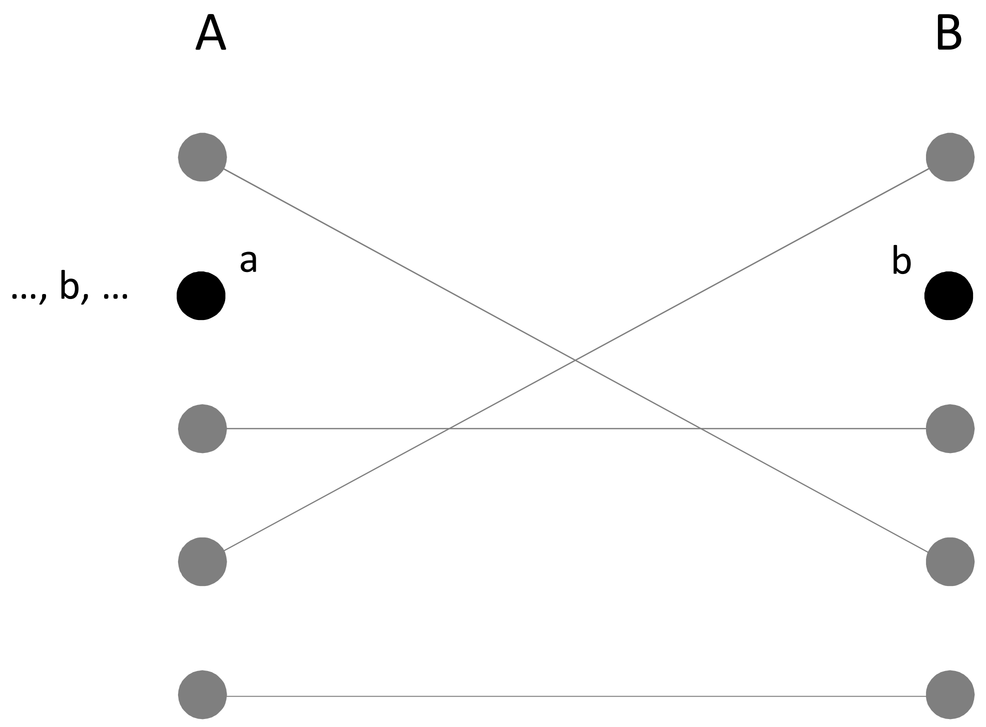
1. Applicants apply for favorite job.
2. Preferred applicants are hired temporarily, rest rejected.
3. Applicants that aren't hired temporarily apply for next choice.
4. Repeat 2 and 3 until everyone is hired.

Gale-Shapley Algorithm



1. Applicants apply for favorite job.
2. Preferred applicants are hired temporarily, rest rejected.
3. Applicants that aren’t hired temporarily apply for next choice.
4. Repeat 2 and 3 until everyone is hired.
5. All applicants are hired permanently.

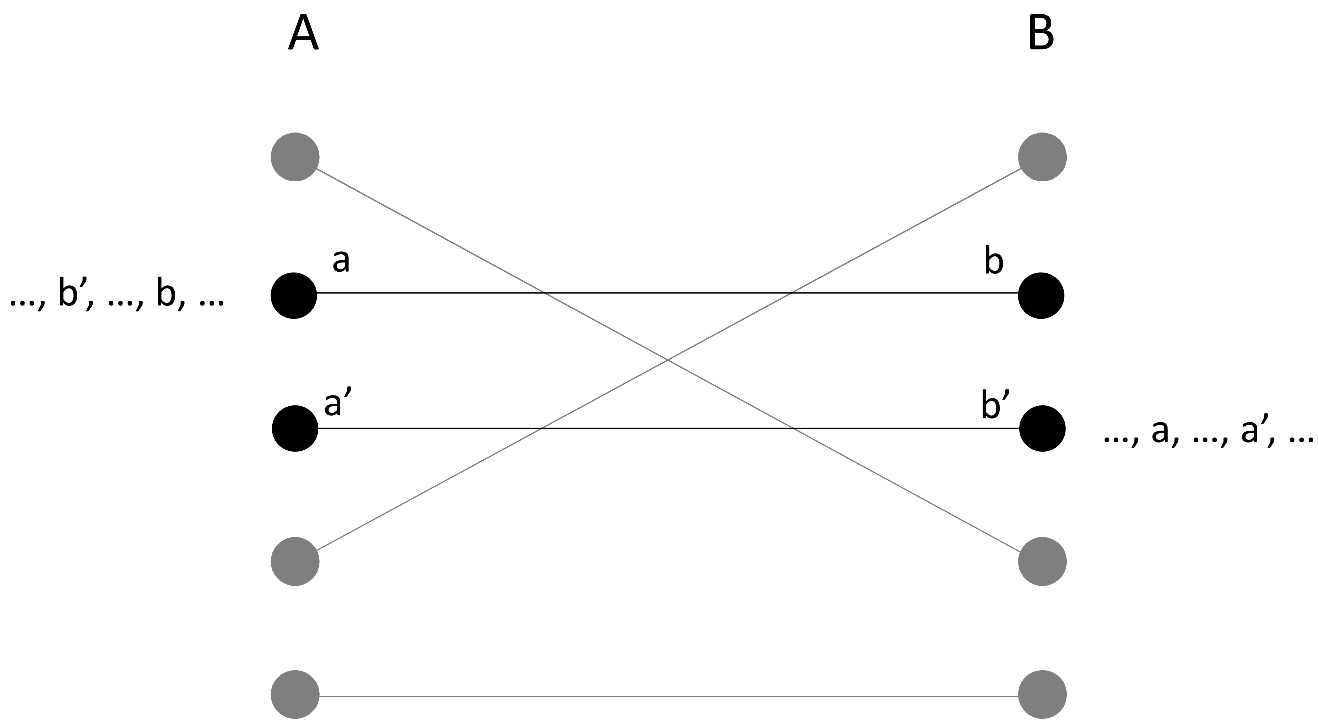
Proof that it produces a perfect matching



Assume the opposite is true:

- If the process is over a must have applied to b.
- From that point on b has to have a temporarily hired applicant.
- Contradiction

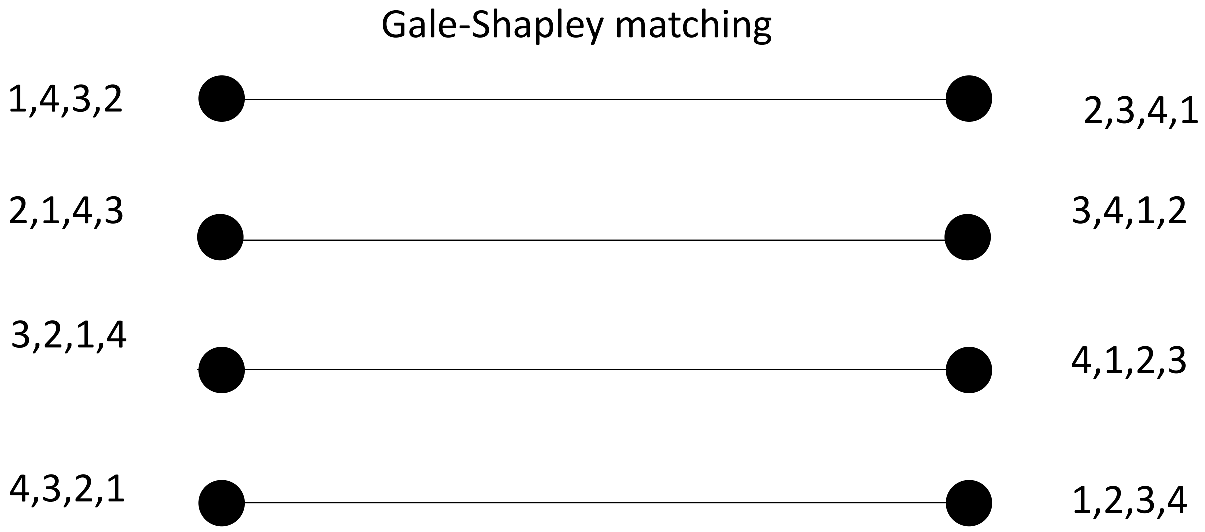
Proof of stability



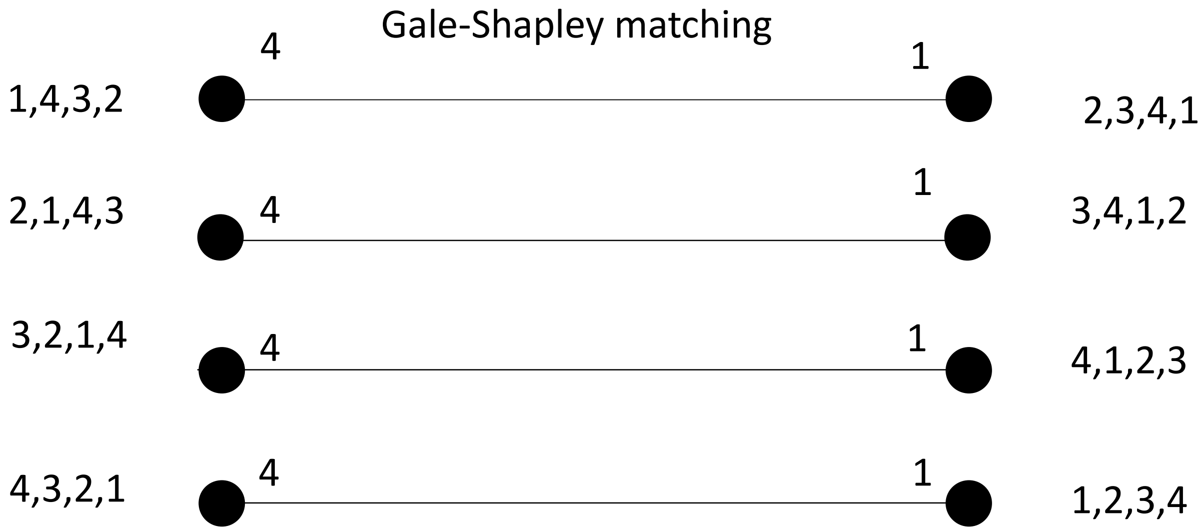
Assume the opposite is true:

- a must have applied to b’ before applying to b.
- Either a was rejected directly or temporarily hired and then rejected. In both cases, b’ must have found a vetex x that he prefers to a.
- If x is not a’ then b’ even prefers a’ to x.
- Contradiction because it would imply that b’ prefers a’ to a.

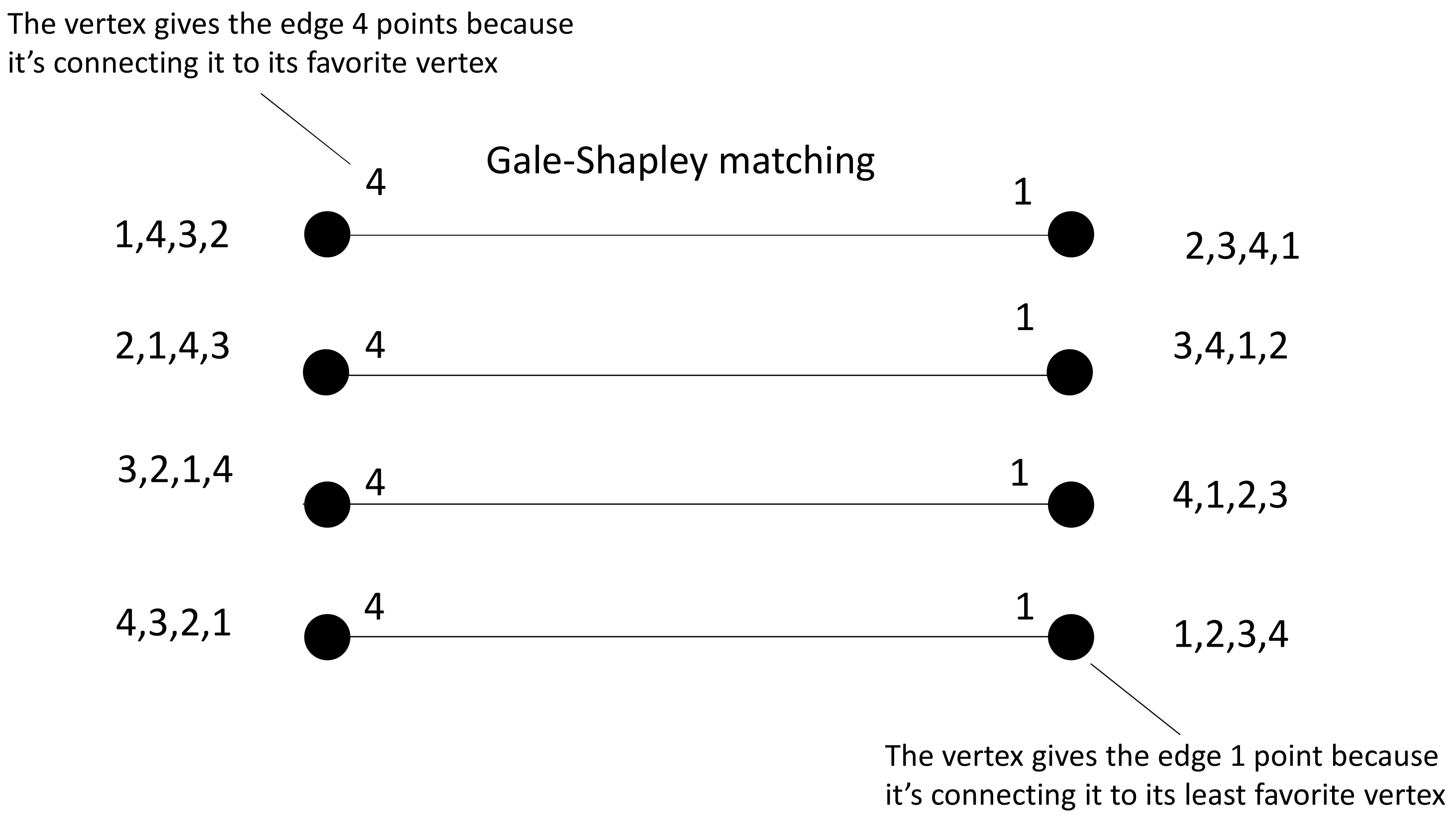
Does the algorithm produce the best matching?



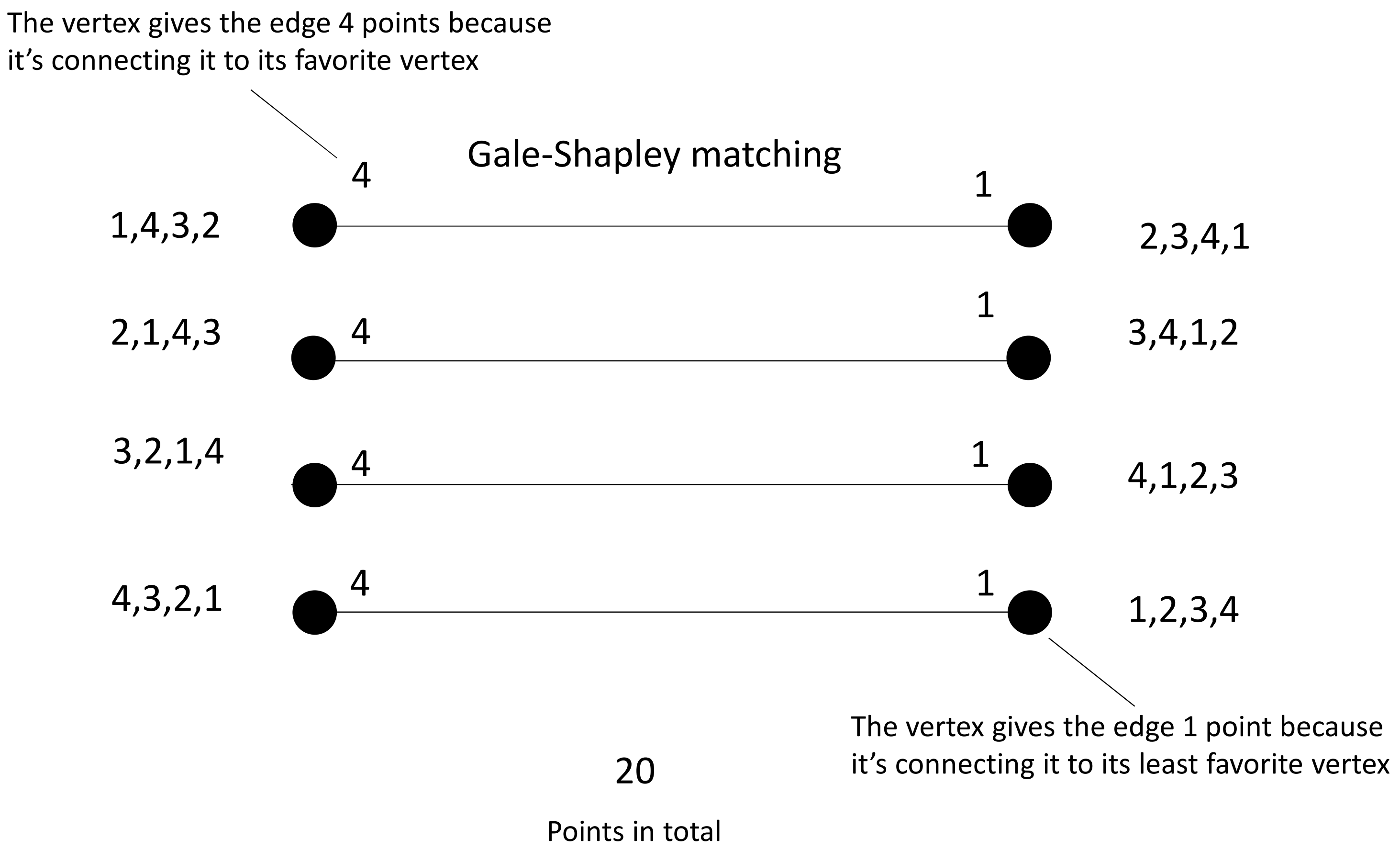
Does the algorithm produce the best matching?



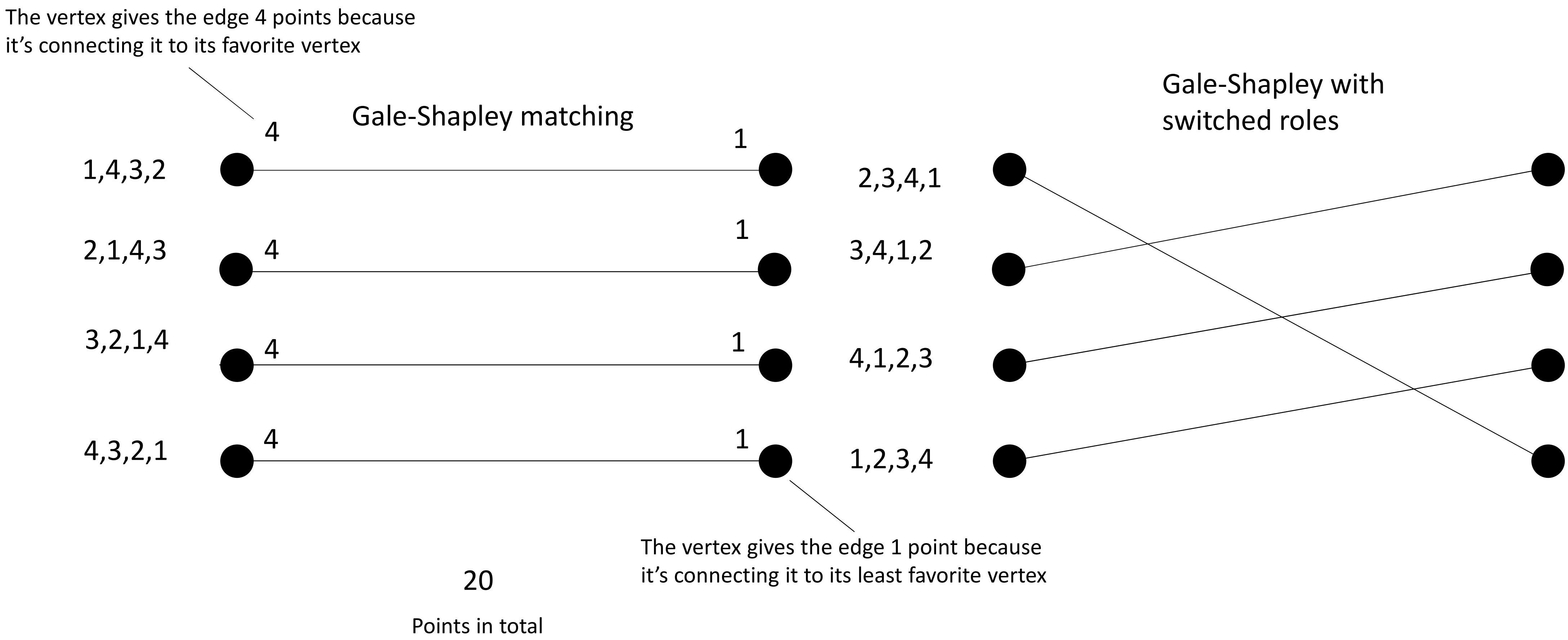
Does the algorithm produce the best matching?



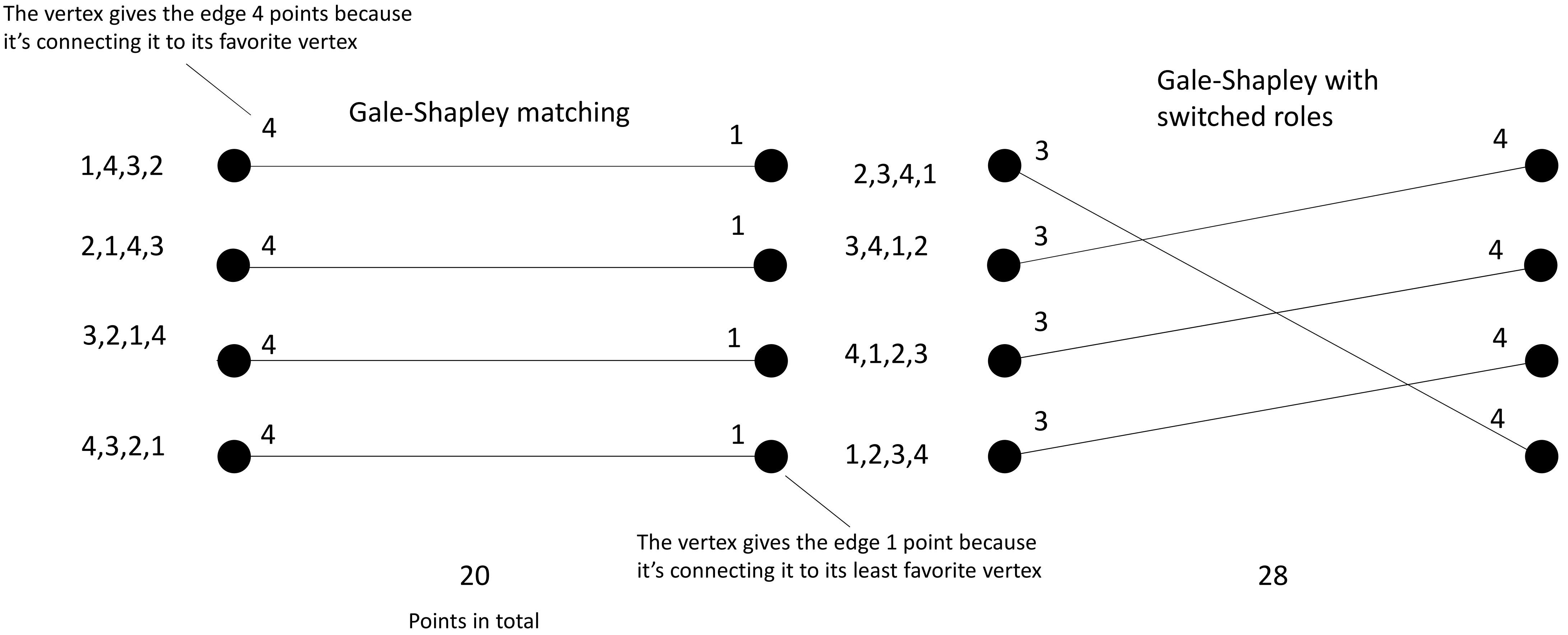
Does the algorithm produce the best matching?



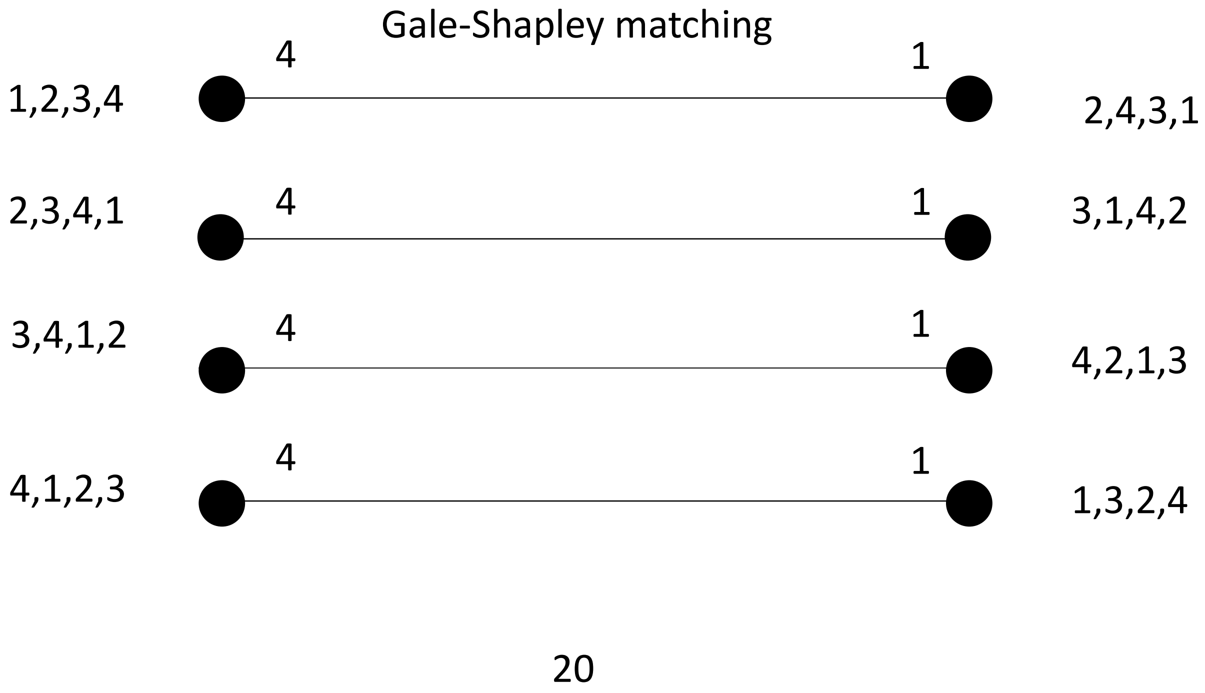
Does the algorithm produce the best matching?

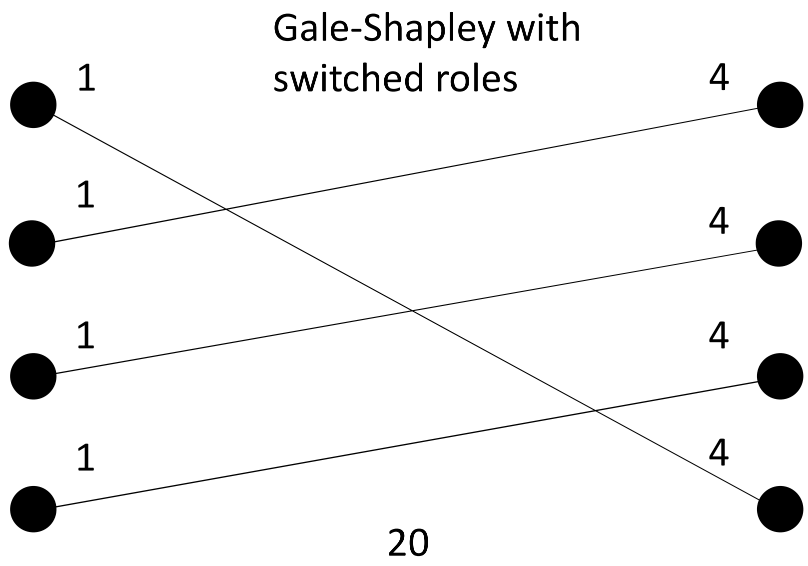


Does the algorithm produce the best matching?

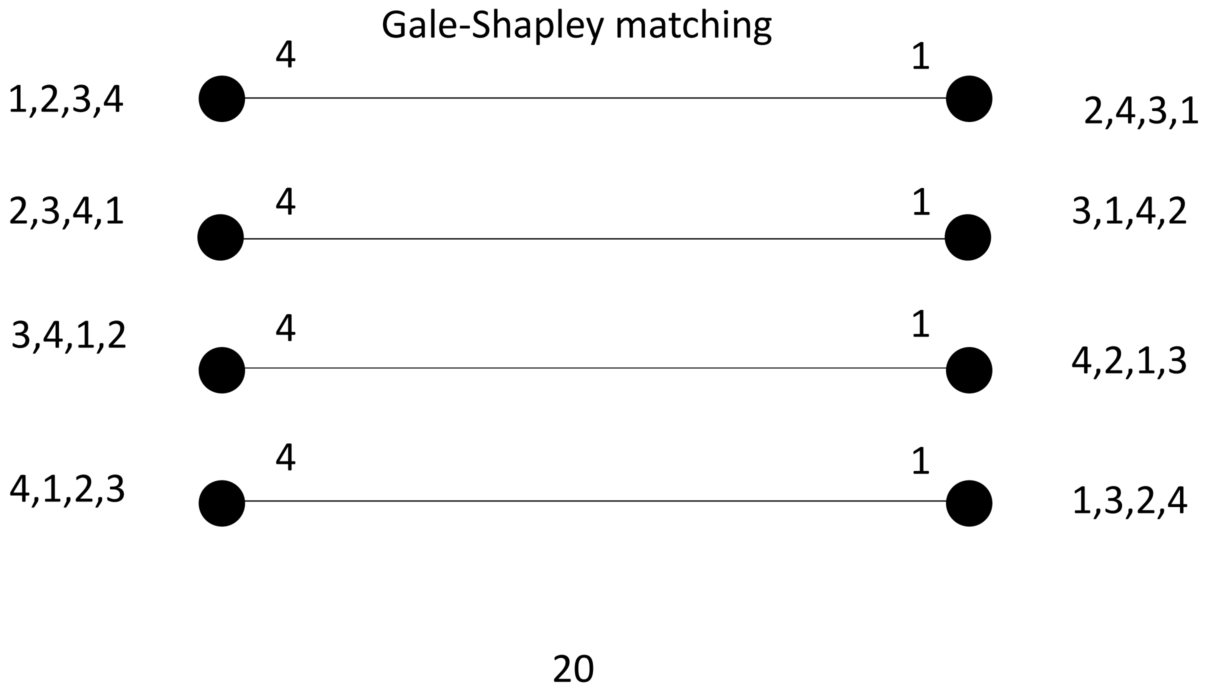


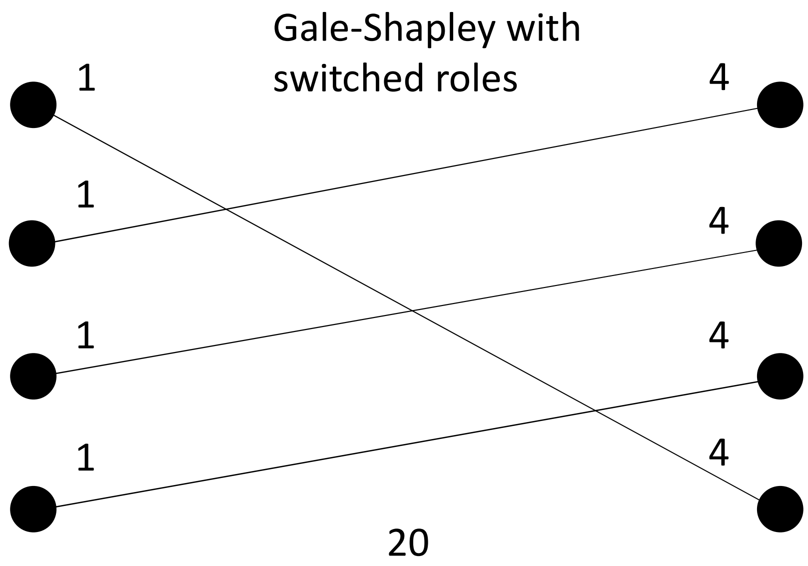
Does running it from both sides produce the best matching?

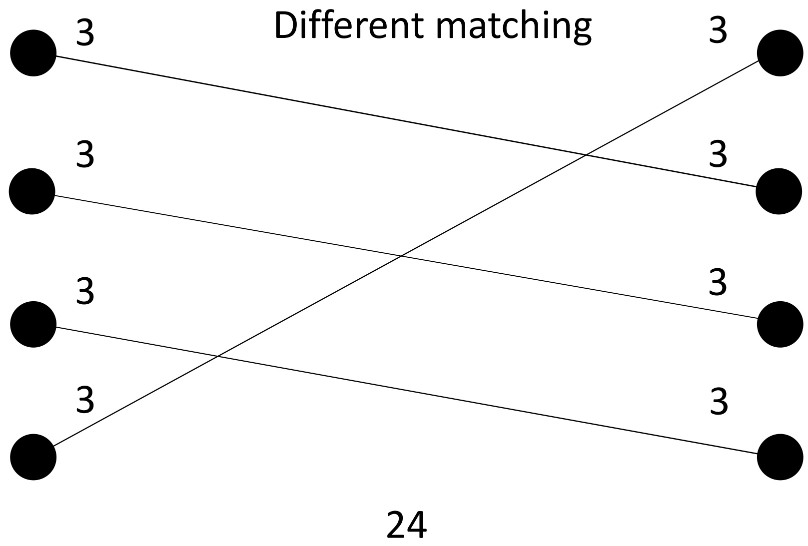




Does running it from both sides produce the best matching?



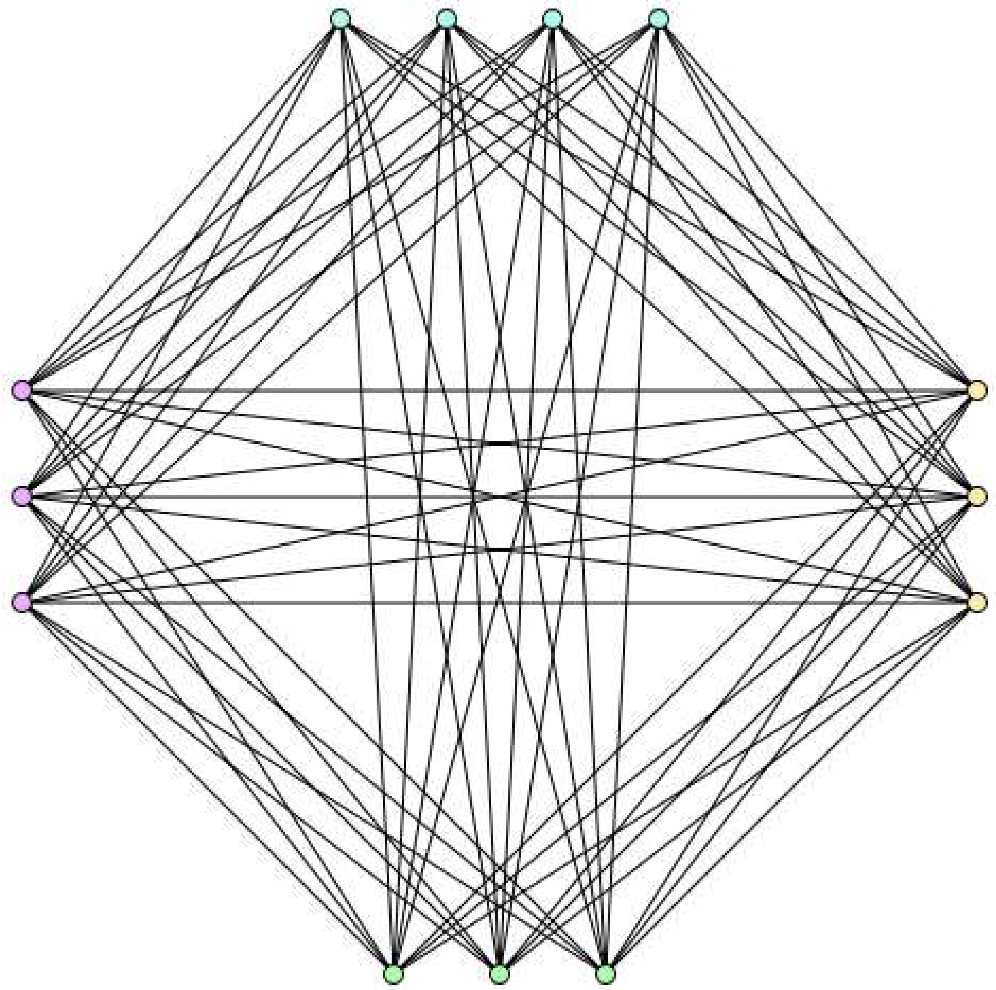




Turan’s Theorem

Karl Robert Kristenprun
Kantonsschule Olten
2ML

Turán Graph



- graph with n-vertices.
- k sets of equal size.
- no two vertices of the same set connected
- doesn’t contain k+1-clique.

- Turán's theorem states that the Turán graph has the largest number of edges among all K_{k+1}-free n-vertex graphs.

Assuming that:

$$\frac{n}{k} \in \mathbb{Z}$$

$$e(T) = \binom{k}{2} \cdot \left(\frac{n}{k}\right)^2 = \left(1 - \frac{1}{k}\right) \cdot \frac{n^2}{2}$$

Let G be maximal K_{k+1}-free graph,

therefore it must contain K_k

If G has $n \leq k$ vertices:

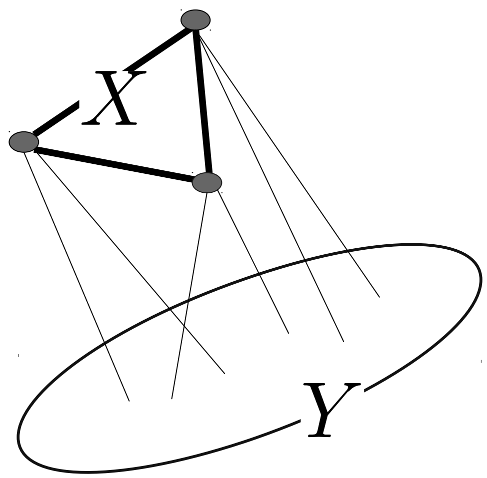
$$\frac{n(n-1)}{2} \leq \left(1 - \frac{1}{k}\right) \frac{n^2}{2}$$

Dividing by n^2 we get:

$$1 - \frac{1}{n} \leq 1 - \frac{1}{k}$$

We divide the graph into two:

- X := K_k
- Y := G \ X



Edges:

$$\binom{k}{2}$$

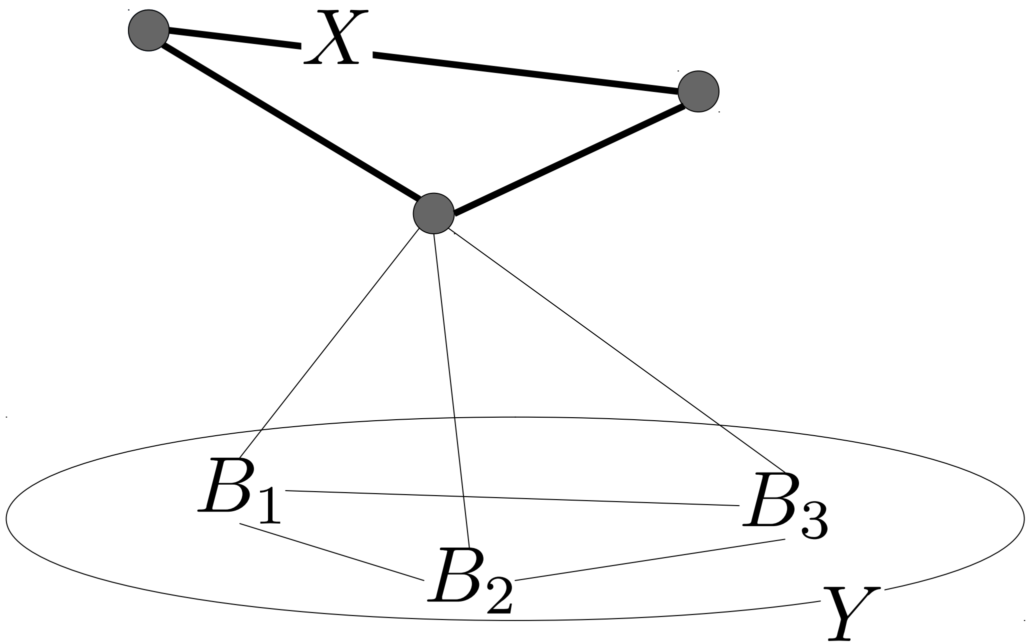
- In X: $\left(1 - \frac{1}{k}\right) \cdot \frac{(n-k)^2}{2}$

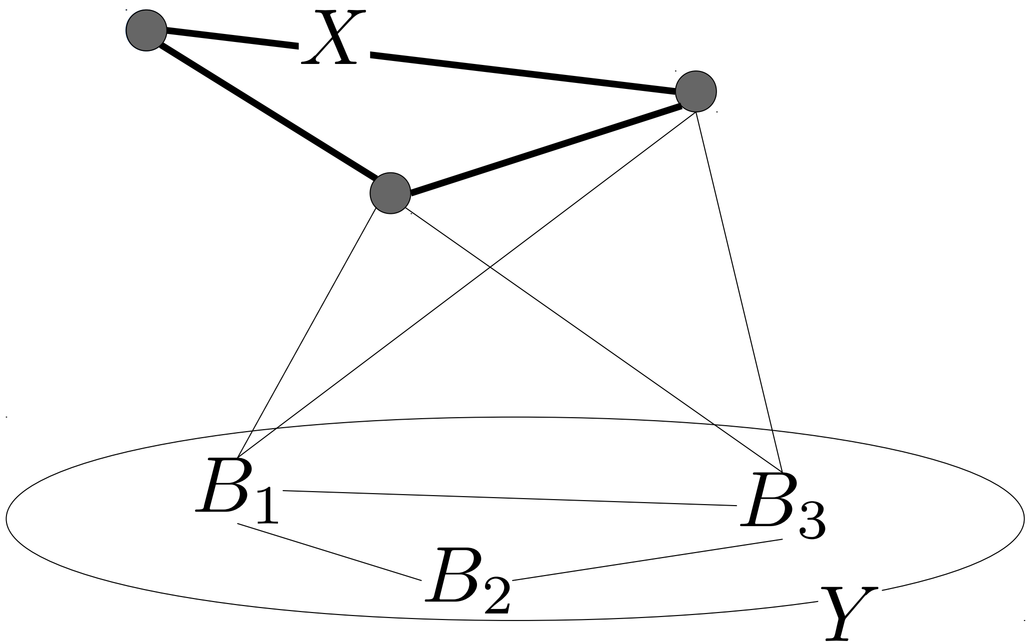
$$(n-k)(k-1)$$

- In Y:
- Between Y and X:

$$\binom{k}{2} + (n-k)(k-1) + \left(1 - \frac{1}{k}\right)\frac{(n-k)^2}{2} =$$

$$= \frac{k-1}{k} \cdot \frac{n^2}{2} = \left(1 - \frac{1}{k}\right)\frac{n^2}{2}$$


X
B_1
B_3
B_2
Y


X
B_1
B_2
B_3
Y

Suppose equality holds:

$$e(G) \leq \left(1 - \frac{1}{k}\right) \frac{n^2}{2}$$