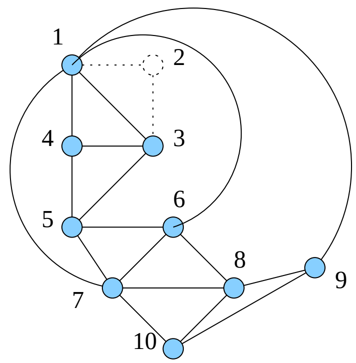
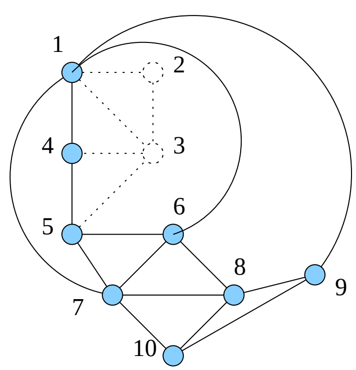
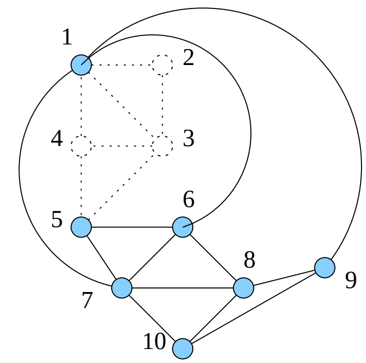


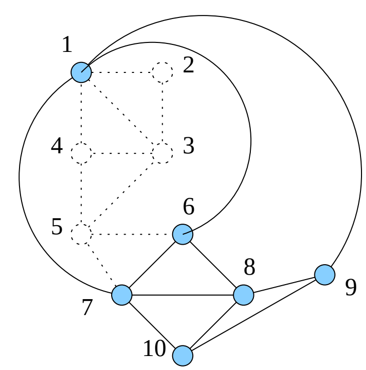
3



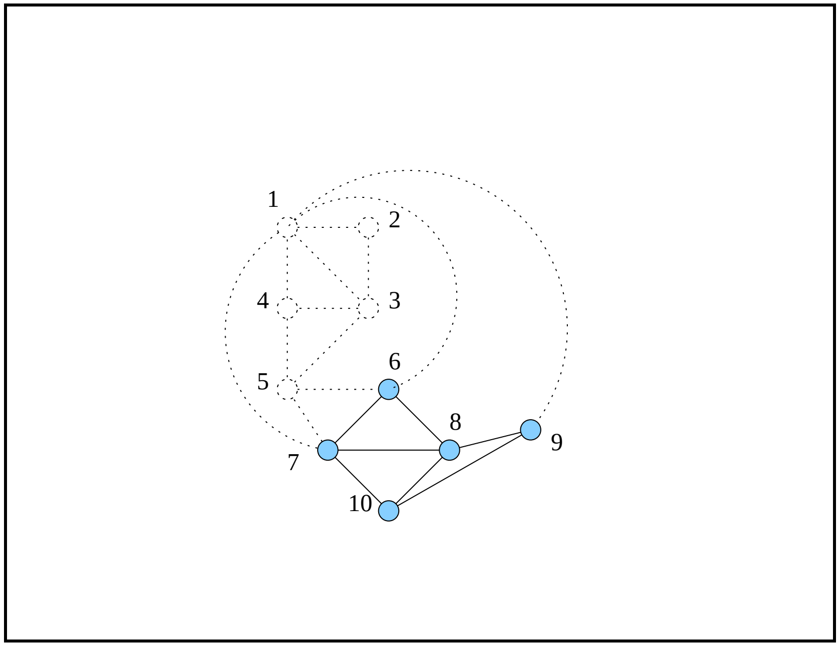
4



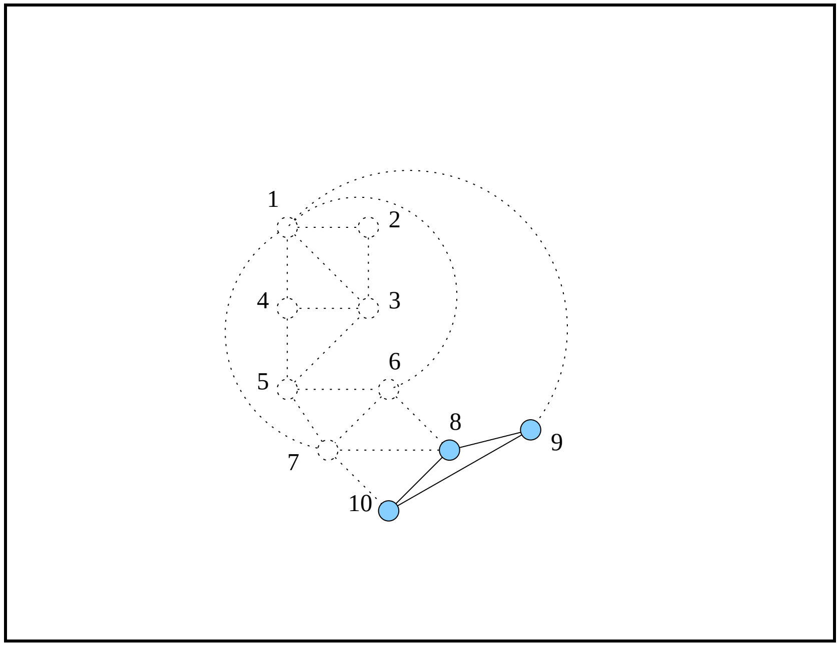
5



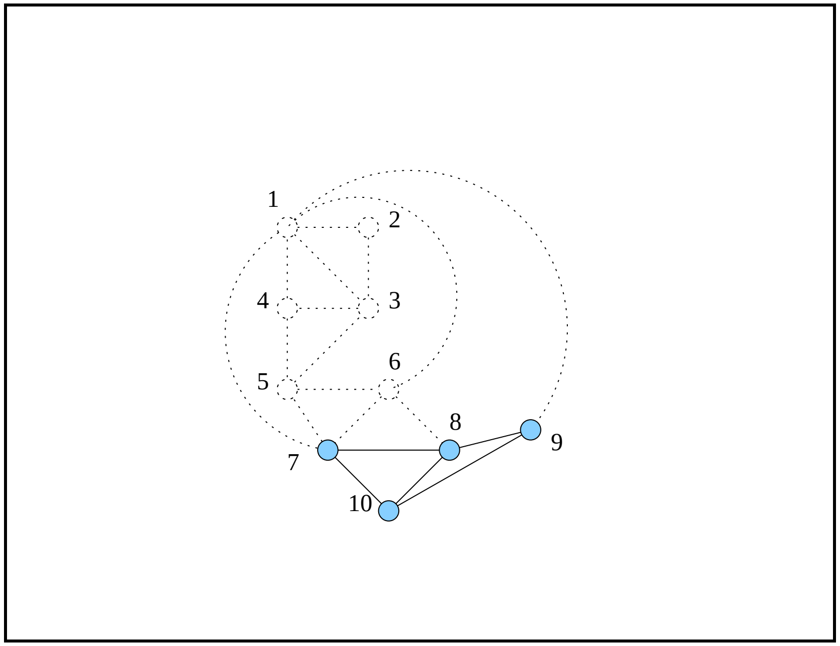
6



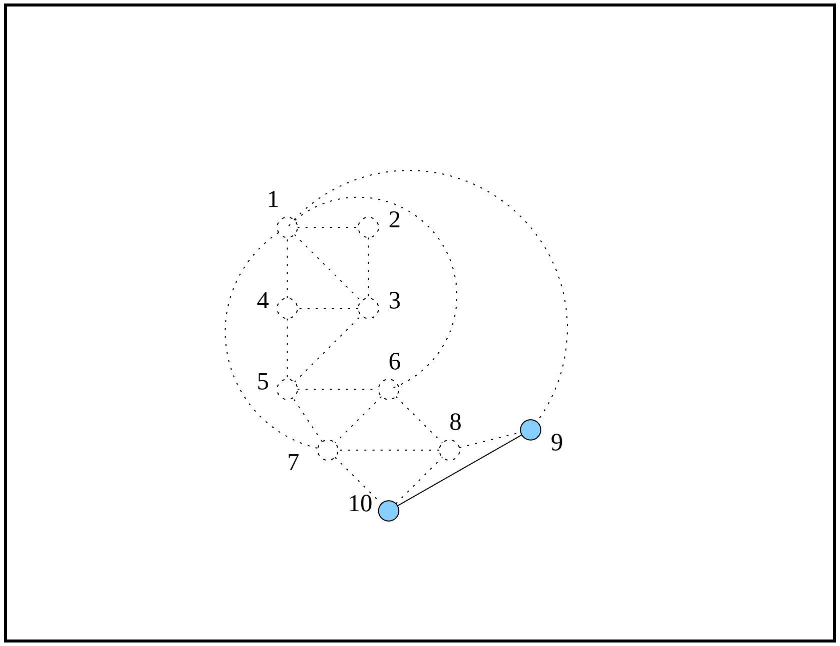
7



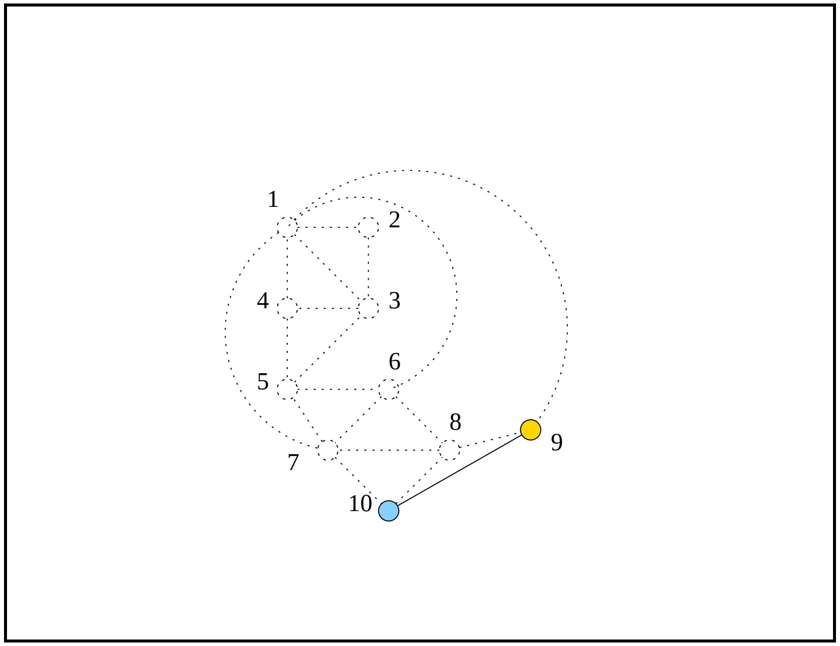
9



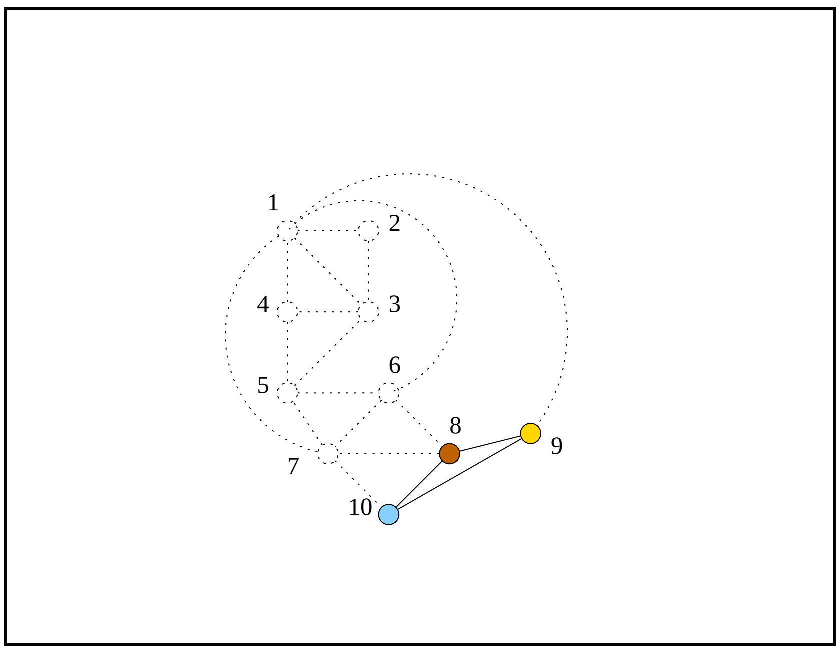
8



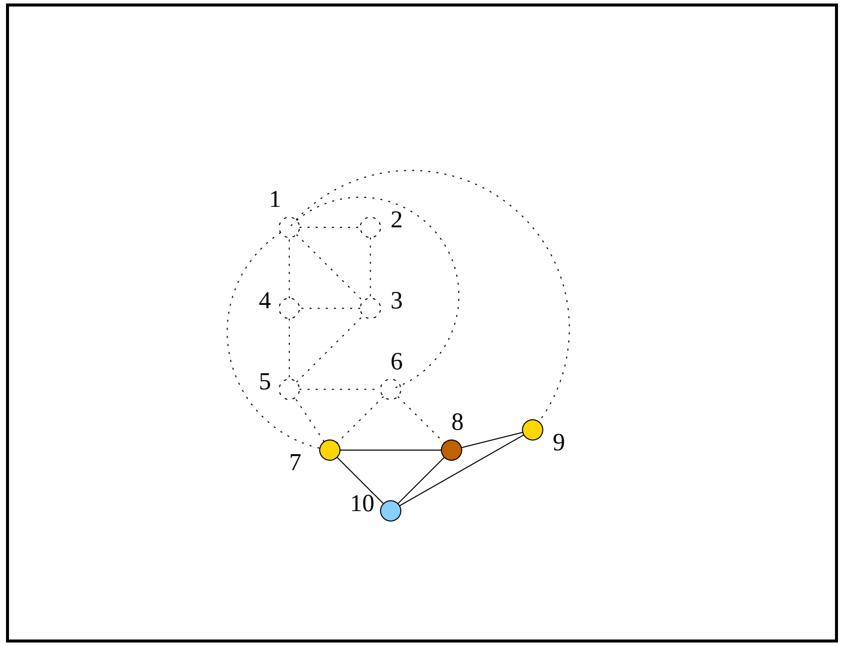
10



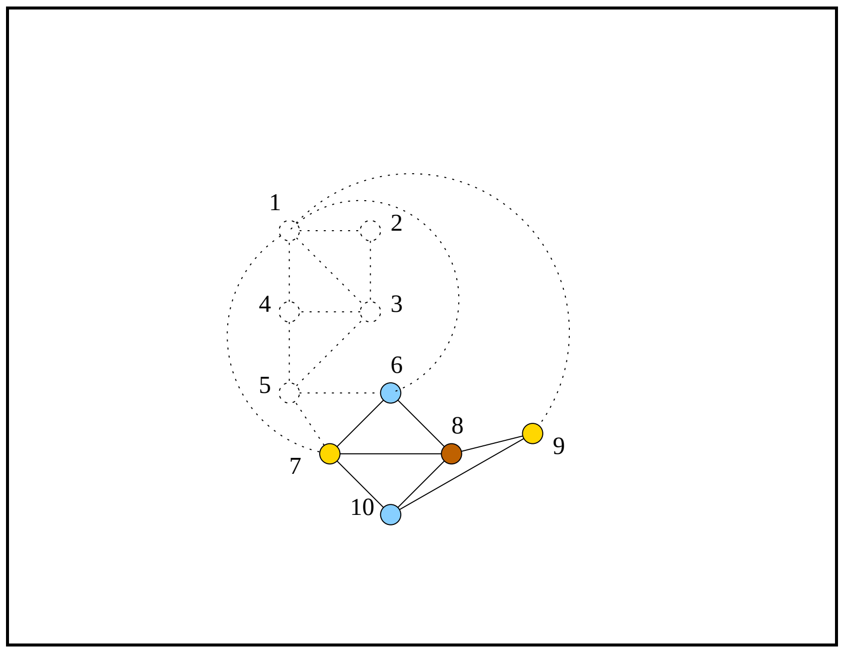
11



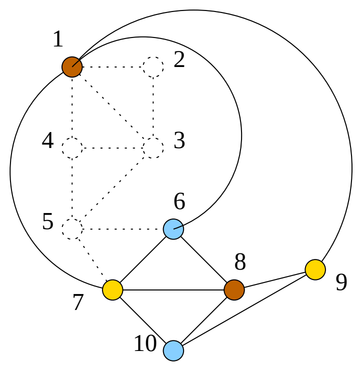
12



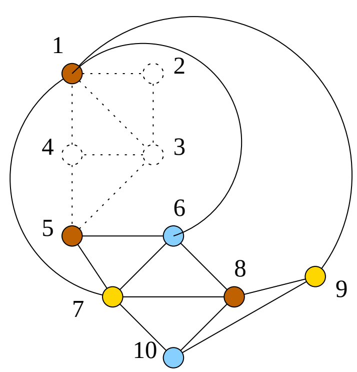
13



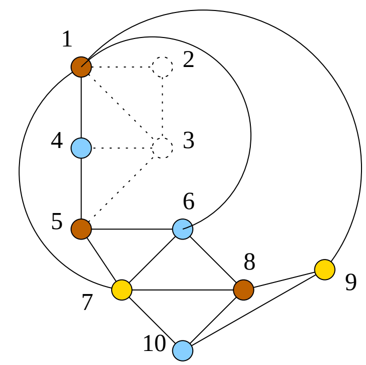
14



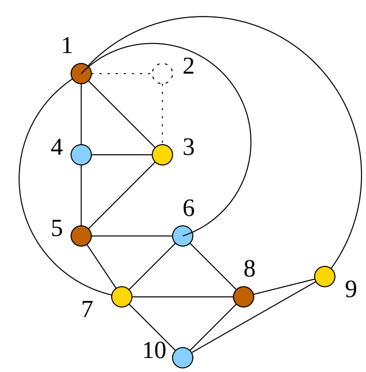
15



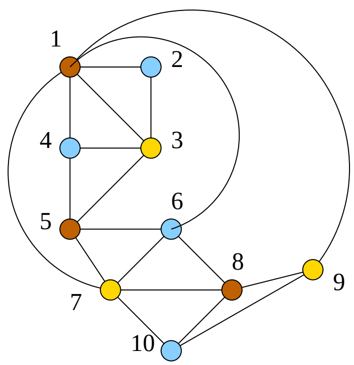
16

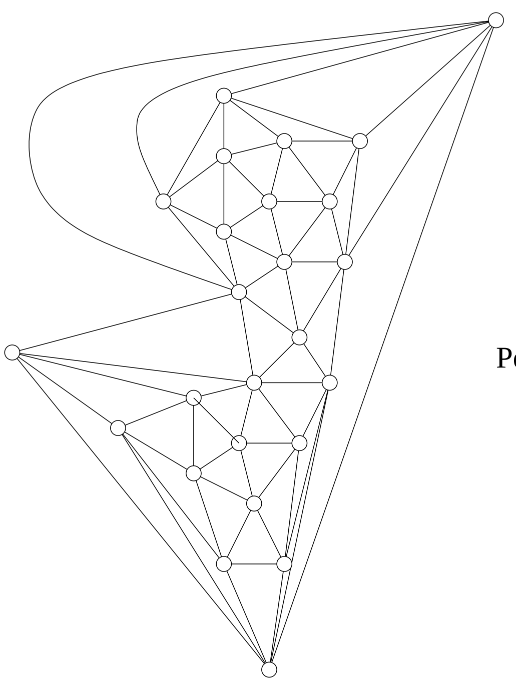


17



18





Percy Heawood exampl

Example 2: edge colouring

We are given a graph and we have to colour its edges with the smallest possible number of colours such that no two adjacent edges have the same colour.

We denote by $\chi' = \chi'(G)$ this number, a.k.a. *chromatic index* of G).

(Vizing) Every graph needs at least Δ and at most $\Delta + 1$ colours to colour its edges.

In fact, the proof of Vizing's statement is based on a polynomial time algorithm that actually finds a colouring using $\Delta + 1$ colours. It is therefore amazing that even a very special case of the edge colouring problem is NP-hard.

(Holyer) The problem of determining the number of colours needed for a 3-regular planar graph is NP-hard.

Putting all this together we can construct another absolute approximation algorithm for an NP-hard optimisation problem. The algorithm A just colours the input graph using $\Delta + 1$ colours as per Vizing's Theorem.

The approximation algorithm A has the performance guarantee $|A_c(G) - \chi'(G)| \leq 1$ for every instance G.

Negative results on absolute approximation

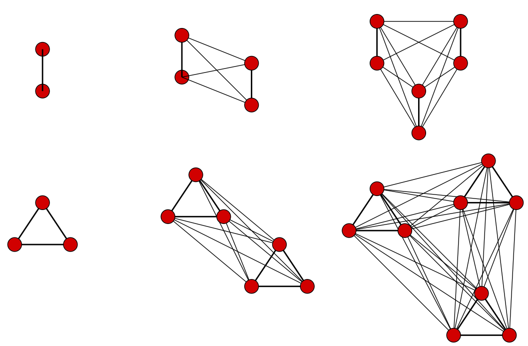
Consider the CLIQUE problem. The problem is that of finding the largest clique (or, complete subgraph) in the input graph G. Let $\omega(G)$ be the size of the largest cliques in G. This is an NP-hard problem [a] The following theorem establishes the hardness of approximating the size of the largest clique in a given graph G.

> If $\mathrm{P} \neq \mathrm{NP}$, then there is no absolute approximation algorithm for the CLIQUE problem.

[a] Note that the problem is essentially (Can you see why MIS and CLIQUE are related?).the same as the MAXIMUM INDEPENDENT SET (MIS) problem.

The mth-power of a graph

Define the m-power of a graph G, say G^m by taking m copies of G and connecting any two vertices that lie in different copies.



Claim. $\omega(G^m) = m \cdot \omega(G)$.

Proof of main theorem

Let us assume for the purposes of contradiction that there is an approximation algorithm A gives an absolute error of k for the clique problem. We claim that the clique problem can be optimally solved by the following strategy.

Run A on G^{k+1}. Let K be the clique returned by the algorithm (we know that $|K| = c(G^{k+1}, A(G^{k+1})))$. Then return the largest projection $K_{\max}$ of K in a copy of G as a proposed solution to CLIQUE in G.

We have that $\omega(G^{k+1}) - c(G^{k+1}, A(G^{k+1})) \leq k$ and hence

$$\omega(G) - \frac{c(G^{k+1}, G^{k+1}))}{k+1} \leq \frac{k}{k+1}$$

Also

$$|K_{\max}| \geq \frac{c(G^{k+1}, A(G^{k+1}))}{k+1}$$

since otherwise $c(G^{k+1}, A(G^{k+1}))$ couldn't be so large. Since both $|K_{\max}|$ and $\omega(G)$ are integer-valued, it follows that $K_{\max}$ must be an optimal clique in G.