Saving an ε: a 2-approximation algorithm for the k-mst problem in graphs

Naveen Garg

Indian Institute of Technology Delhi

Presented by: Julian Mestre, University of Maryland, College Park

The k-MST problem

- Given a weighted graph and a number k we want to find a tree spanning k vertices and having the minimum possible weight.
- The problem is NP-hard and max-SNP-hard.
- This paper presents a 2-approximation algorithm for this problem

Previous Work

There has been much work on the problem in the last decade.

year	authors	approx ratio
1993	Ravi et.al.	O($k^{1/2}$)
1995	Awerbuch, Azar, Blum, Vempala	O($\log^2 k$)
1995	Rajagopalan, Vazirani	O($\log k$)
1996	Blum, Ravi, Vempala	17
1996	Garg	3
1998	Arya and Ramesh	2.5
2000	Arora and Karakostas	$2+\varepsilon$

k-MST on the plane

The Euclidean version of the problem is also very well studied.

year	authors	approx ratio
1993	Ravi et.al.	O($k^{1/4}$)
1994	Garg, Hochbaum	O($\log k$)
1995	Eppstein	O($\log k / \log\log n$)
1995	Blum, Chalasani, Vempala	O(1)
1999	Mitchell, Blum, Chalasani, Vempala	2
1998	Arora	$1+\varepsilon$

The ideas

- As in previous work on constant factor approximations for this problem, our algorithm relies on the Goemans-Williamson procedure for the prize collecting Steiner tree problem.
- We show how to compute, in $O(mn^3 \log n)$ time, a threshold potential, at which the size of the largest tree in the forest, crosses k.
- Our key theorem is to show that we can pick a tree of cost at most twice a suitable bound.
- By running our procedure on n different subgraphs and picking the best tree found we can argue a 2-approximation.

Growth Phase of modified GW

- Each vertex in V is assigned an initial potential p.
- Initially, the set of tight edges F, is empty.
- A component of (V, F) with positive residual potential is called *active*.
- We increase the dual variable y_S for every active set S, simultaneously decreasing its residual potential.
- This is done till some edge goes tight (total dual across edge equals its cost). Let e be an edge between sets S_1, S_2 that went tight.
- $F \leftarrow F \cup \{e\}$. S_1, S_2 are merged into one set S whose residual potential is the sum of the residual potentials of S_1, S_2.
- Process continues till all components become inactive.

modified GW: illustrated

Each vertex has an initial potential of 3.

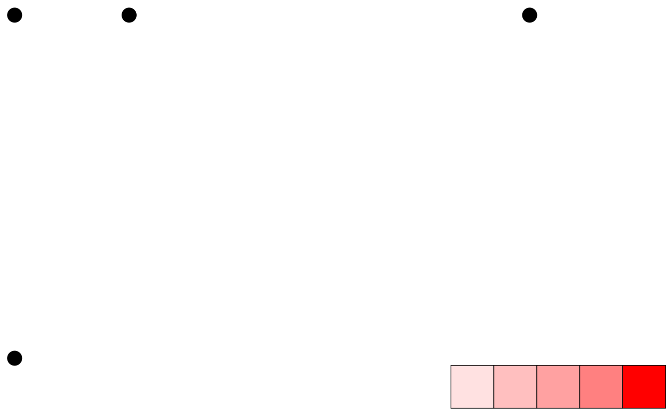


Figure 1: At time 0 unit

modified GW: illustrated

Each vertex has an initial potential of 3.

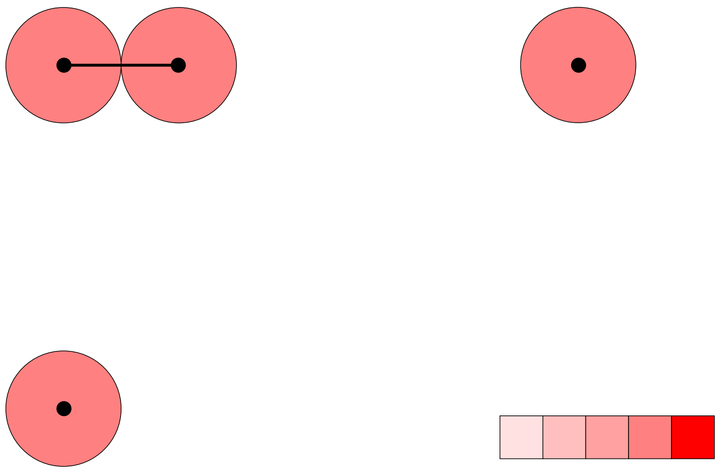


Figure 1: At time 1 unit

modified GW: illustrated

Each vertex has an initial potential of 3.

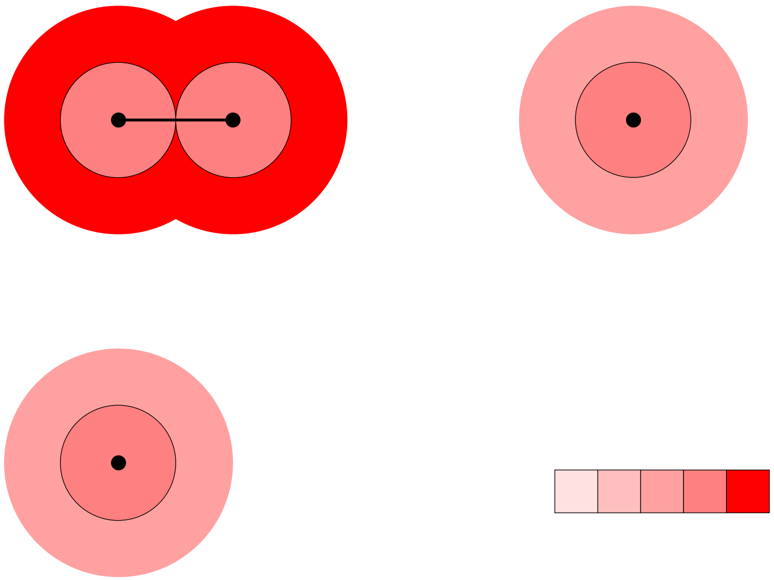


Figure 1: At time 2 unit

modified GW: illustrated

Each vertex has an initial potential of 3.

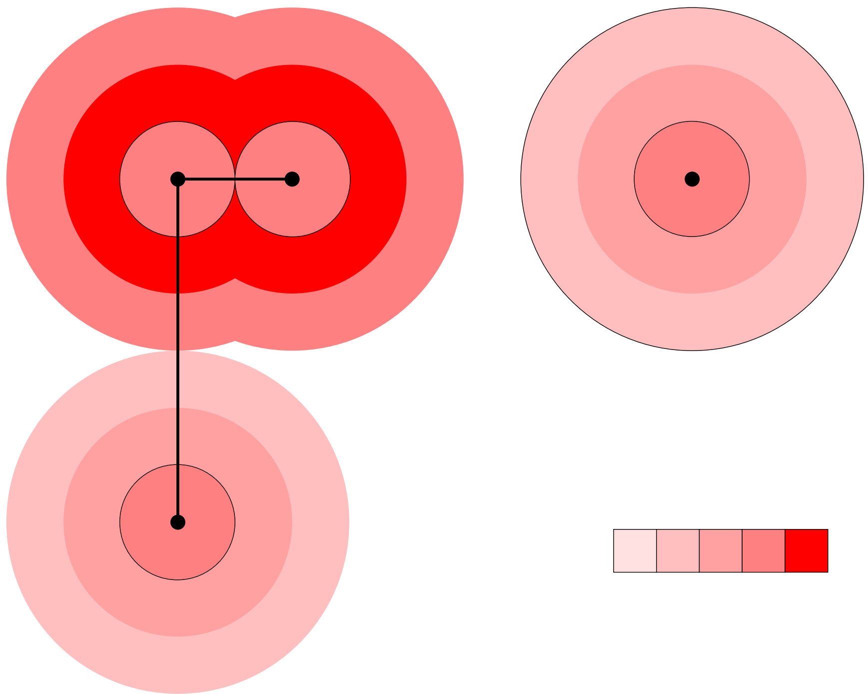


Figure 1: At time 3 unit

modified GW: illustrated

Each vertex has an initial potential of 3.

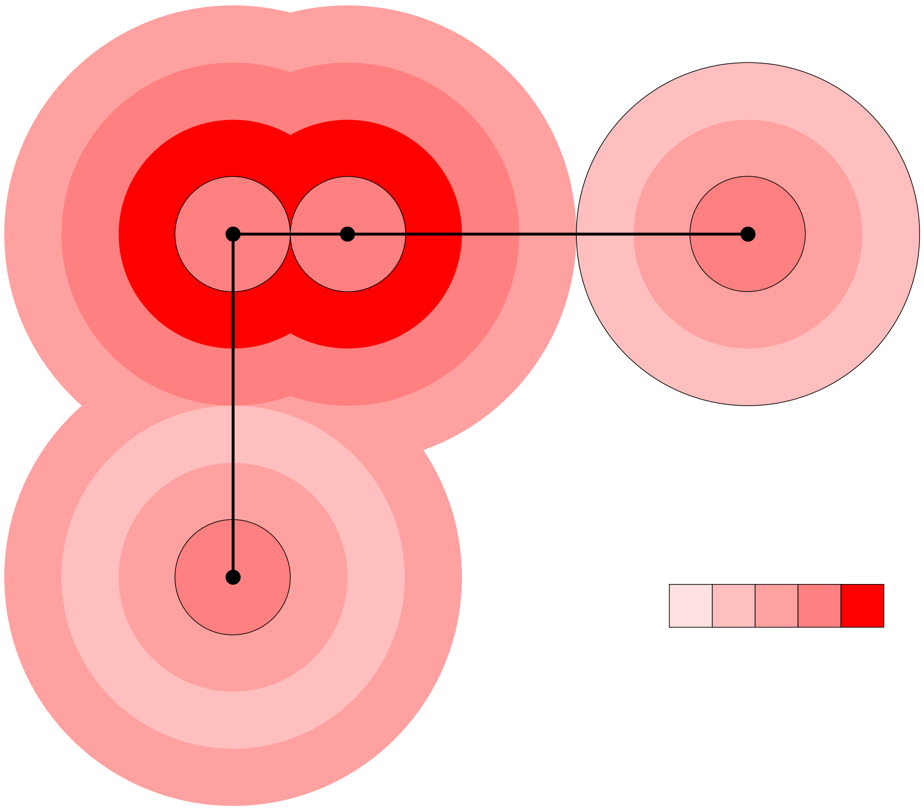


Figure 1: At time 4 unit

modified GW: illustrated

Each vertex has an initial potential of 3.

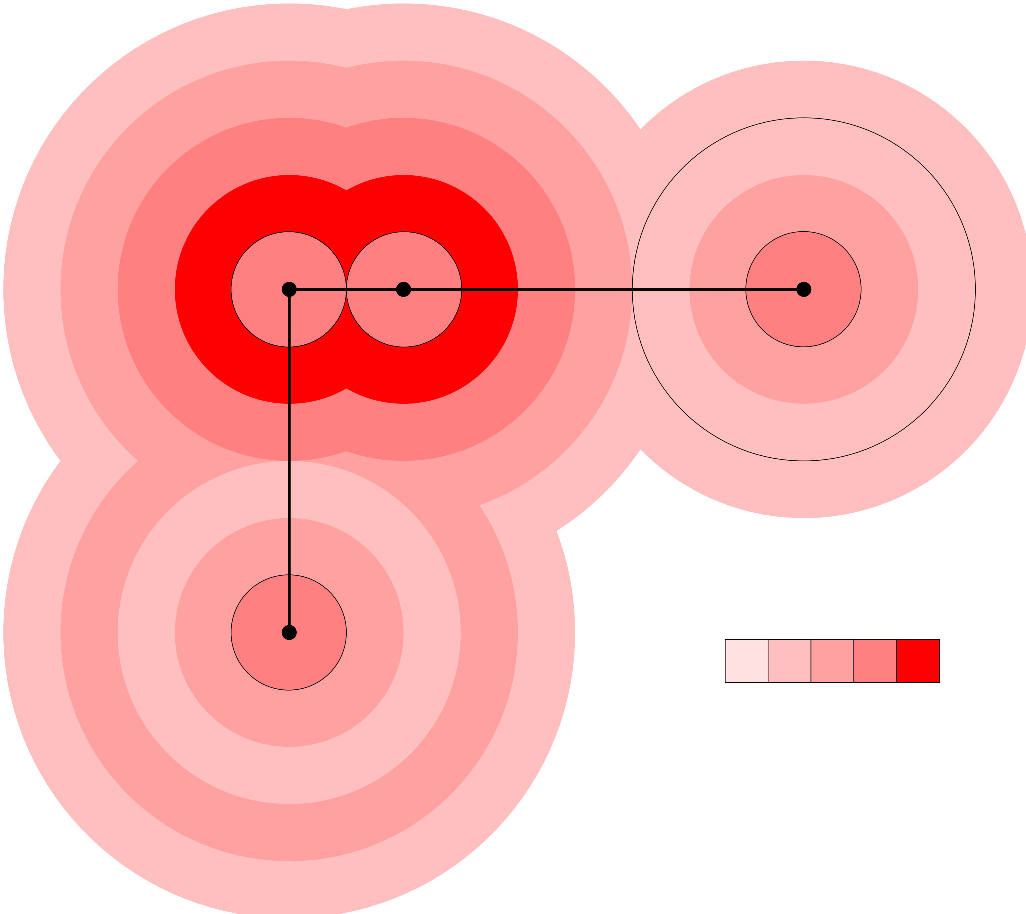


Figure 1: At time 5 unit

Some Notation

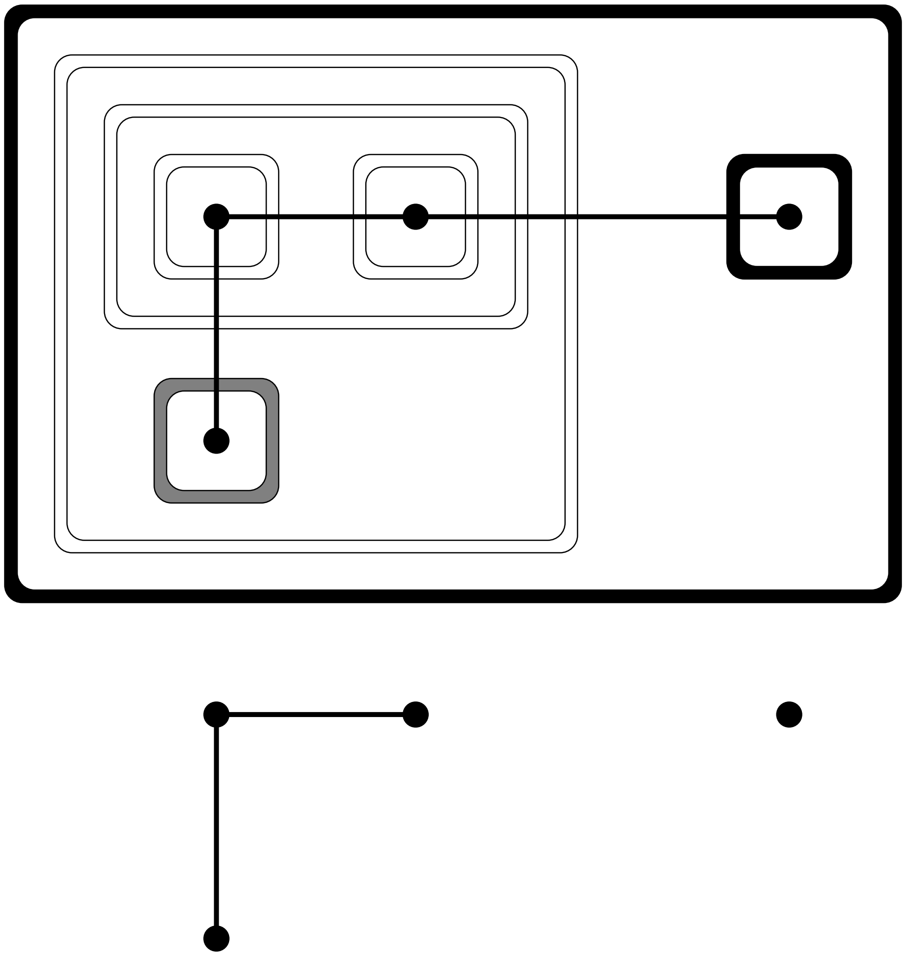
- F_p is the forest obtained after the growth phase.
- $\mathcal{S}_p$ is the collection of sets S such that $y_S > 0$.
- $\chi_p : \mathcal{S}_p \longrightarrow \{black, grey, white\}$ is a coloring function defined as follows

$S \in \mathcal{S}_p$ is black if S is an inactive component at any point, it is colored grey if S becomes inactive just as it merges with another set and S is colored white otherwise.

The Pruning phase

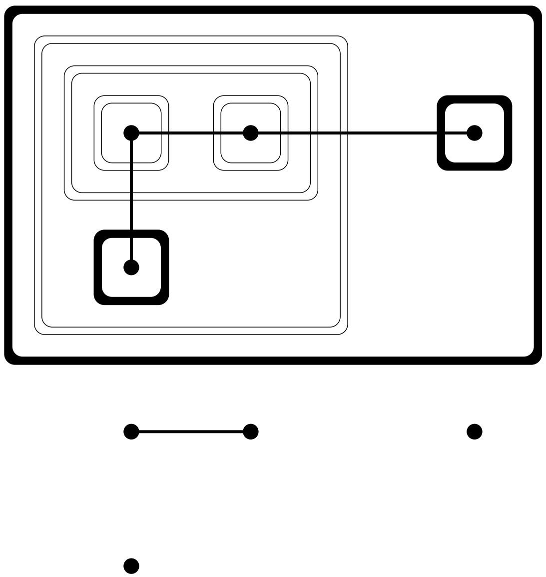
- An edge e is pruned (dropped from F_p) if it is the only edge in F_p which is incident to a black set.
- $\widehat{F_p}$ is the set of edges that remain after pruning.
- Let $\alpha(p)$ be the size of the largest tree in $\widehat{F_p}$.
- q is a threshold potential if $\alpha(q-) < k$ and $\alpha(q+) > k$ where $q-, q+$ are potentials which are infinitesimally smaller and larger than q.

$\mathcal{S}_p, \chi_p, F_p$ and $\widehat{F_p}$ illustrated



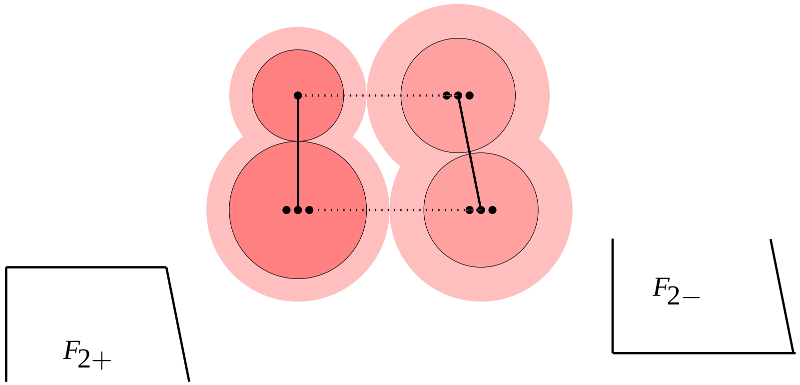
$\mathcal{S}_p, \chi_{p-}, F_{p-}$ and $\widehat{F_{p-}}$ illustrated

If in the previous example the initial potential was less than 3 then the set which was grey earlier would be colored black and this would cause an additional edge to be pruned.



F_{p-} and F_{p+} can be very different

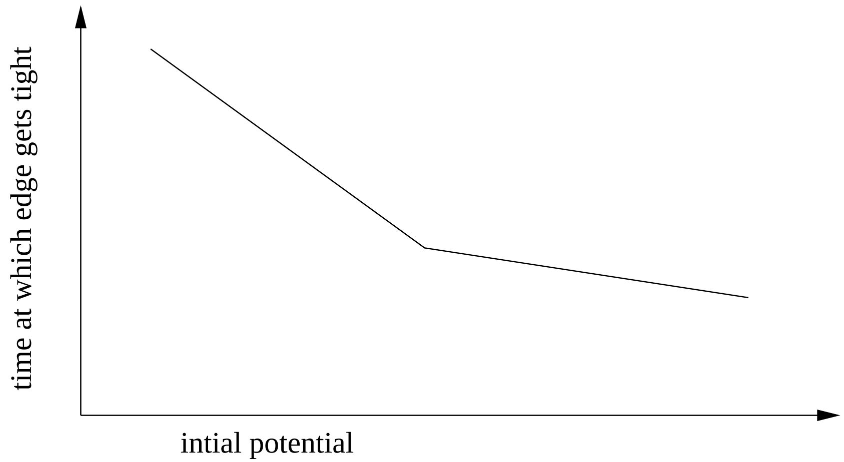
- Infinitesimal changes in initial potential can cause the forest of tight edges to change considerably.



- Increase in initial potential causes the top-left set to become inactive a little later. This results in the top horizontal edge getting tight before the lower one.
- Effect is reversed when initial potential is lowered.

Plotting effect of tiny changes

- A plot of the time at which the edge gets tight with change in initial potential of the vertices looks like



- This however assumes that the set of tight edges and the color of the sets remains unchanged as the initial potential is varied.
- But we can still use this procedure for computing the threshold potential.

Computing the threshold potential

- Let l, r be initial potentials such that $\alpha(l+) < k \leq \alpha(r-)$.
- Further l, r are such that the runs of $\texttt{modGW}(l+)$ and $\texttt{modGW}(r-)$ agree on the first i edges.
- We can either find a threshold potential or narrow the range (l, r) to (l', r') such that $\alpha(l'+) < k \leq \alpha(r'-)$ and the the runs of $\texttt{modGW}(l'+)$ and $\texttt{modGW}(r'-)$ agree on the first $i+1$ edges.
- Thus within $n-1$ runs of this procedure we will find a threshold potential.

The key theorem

Let q be a threshold potential. We can find a tree spanning k vertices and having cost at most $2(qk - \pi(Q))$ where Q is any set in $\mathcal{S}_q$ containing all of T and $\pi(Q)$ is the potential of Q when it was formed.

We would take Q as that set in $\mathcal{S}_q$ which contains all of T and has maximum $\pi()$.

Using the theorem

Lower Bound: Let T^* be the optimum tree and $O \in \mathcal{S}_q$ be the minimal set including all of T^*. Then the optimum solution has cost at least $qk - \pi(O)$.

Thus if $\pi(O) \leq \pi(Q)$ we are done since

$$cost(T) \leq 2(qk - \pi(Q)) < 2(qk - \pi(O)) \leq 2 \cdot cost(T^*)$$

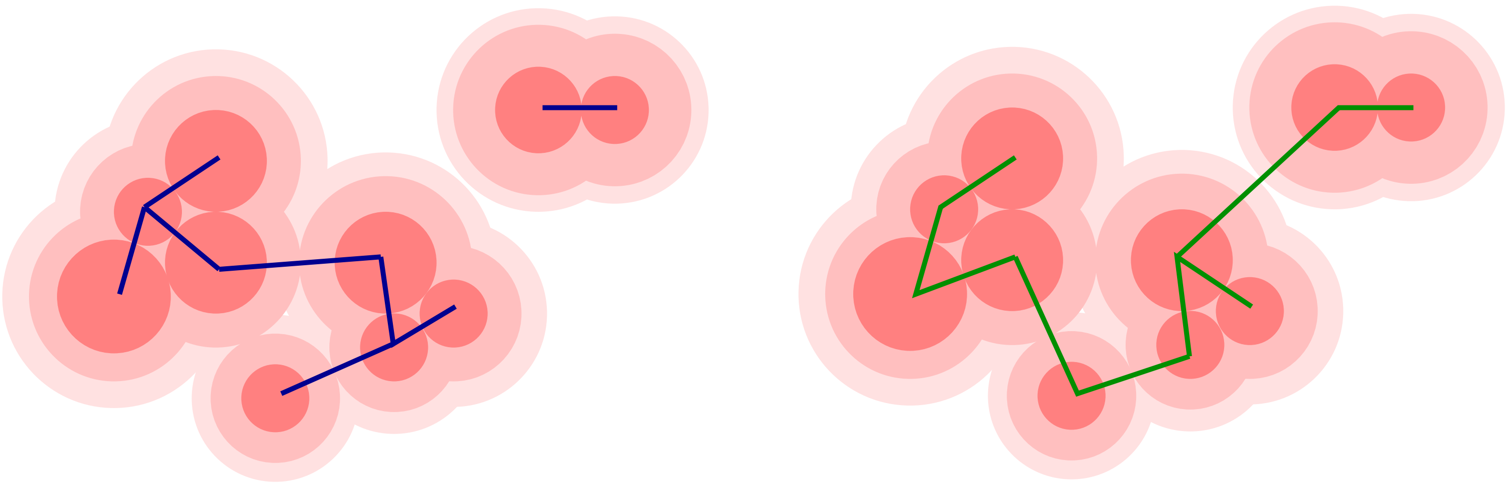
Using the theorem (contd.)

- Let M be a maximal collection of sets in $\mathcal{S}_q$ which have initial potential strictly larger than $\pi(Q)$.
- If $\pi(O) > \pi(Q)$ then all of T^* is contained in some set of M.
- We find 2-approximations to the k-mst in the subgraph induced by the vertices of each set in M.
- The best of these $M + 1$ trees is then a 2-approximation to the optimum.
- Since the sets in M are disjoint and no set contains all vertices of T, the number of times we will have to compute k-trees is at most n.

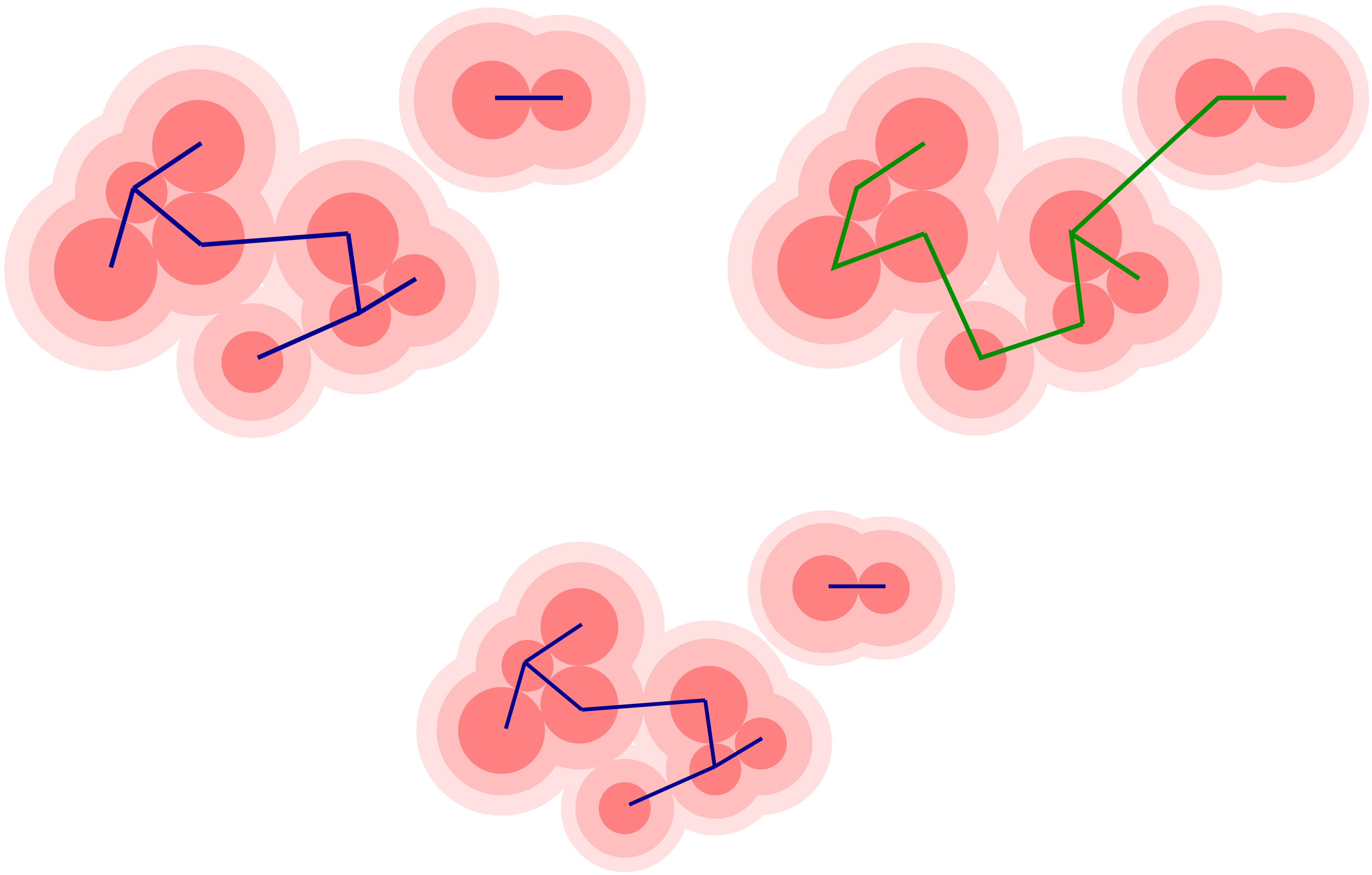
Proving the key theorem

- Let q be the threshold potential
- The forests F_{q-} and F_{q+} can be vastly different.
- However, the vertices of every set in $\mathcal{S}_q$ are contiguous in both the forests.
- We consider the sets in $\mathcal{S}_q$ in the order they were created and replace the subtree induced in F_{q-} with the subtree in F_{q+} by replacing one edge at a time

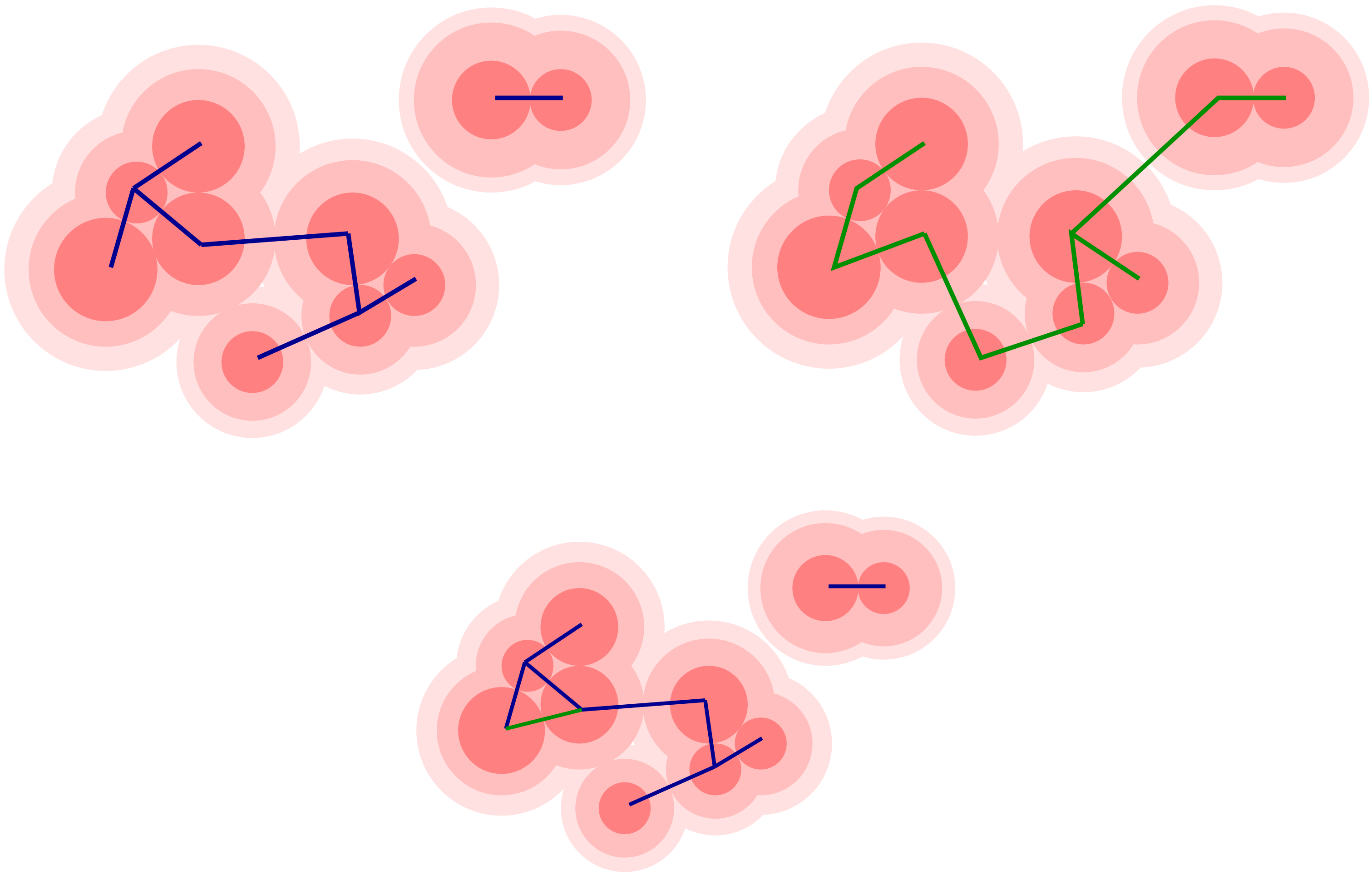
moving from F_{q-} to F_{q+}



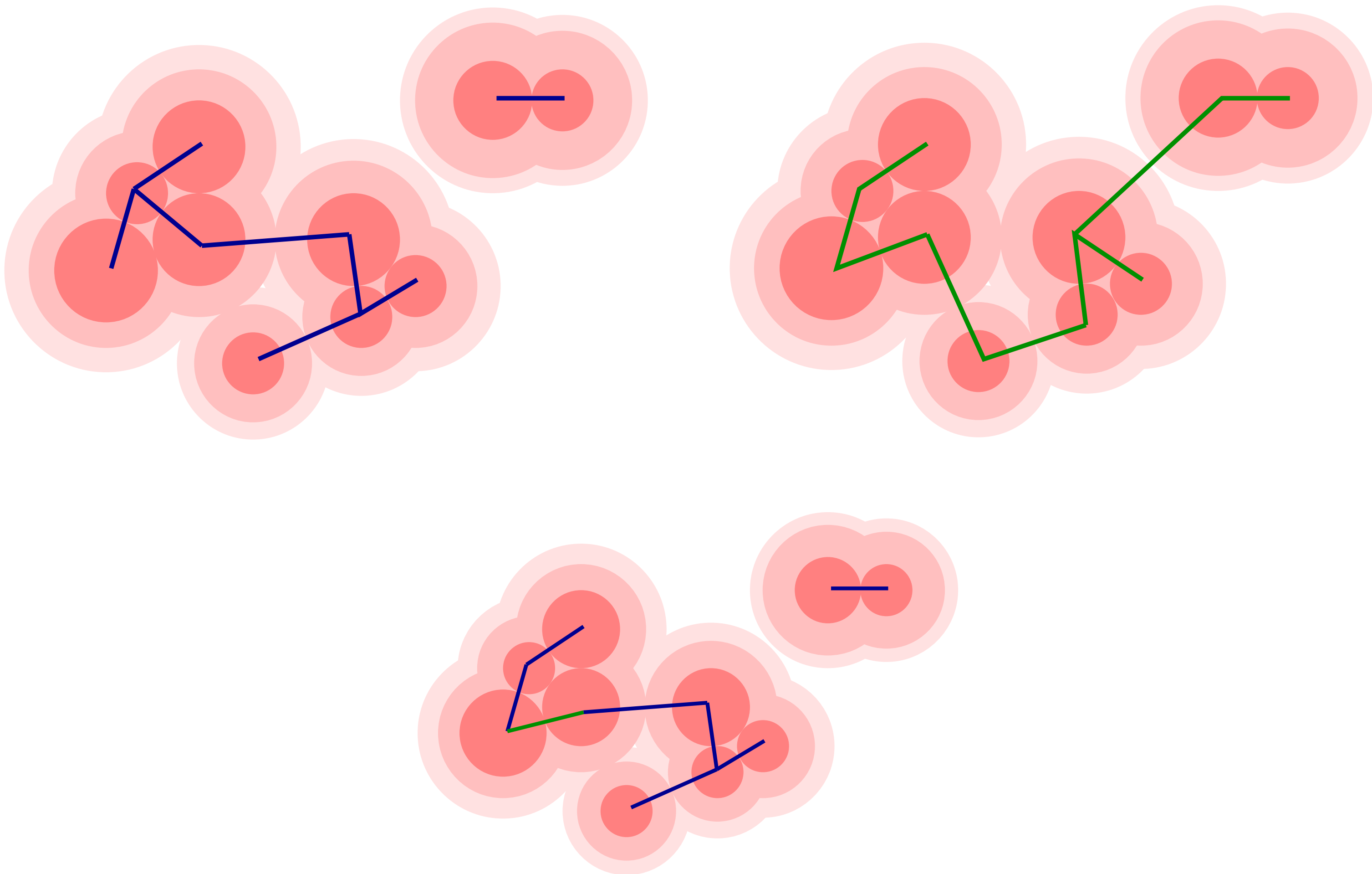
moving from F_{q-} *to* F_{q+}



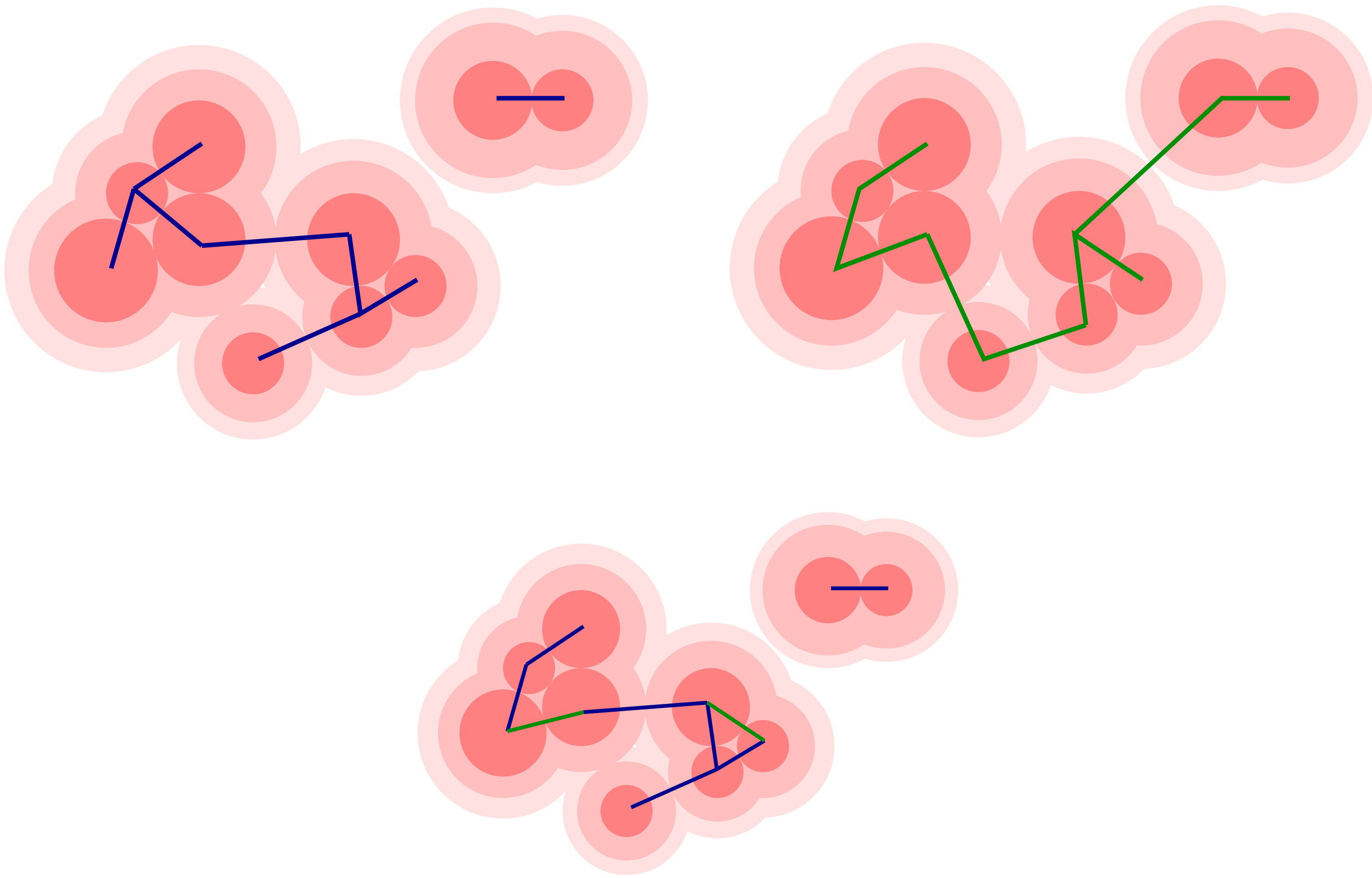
moving from F_{q-} to F_{q+}



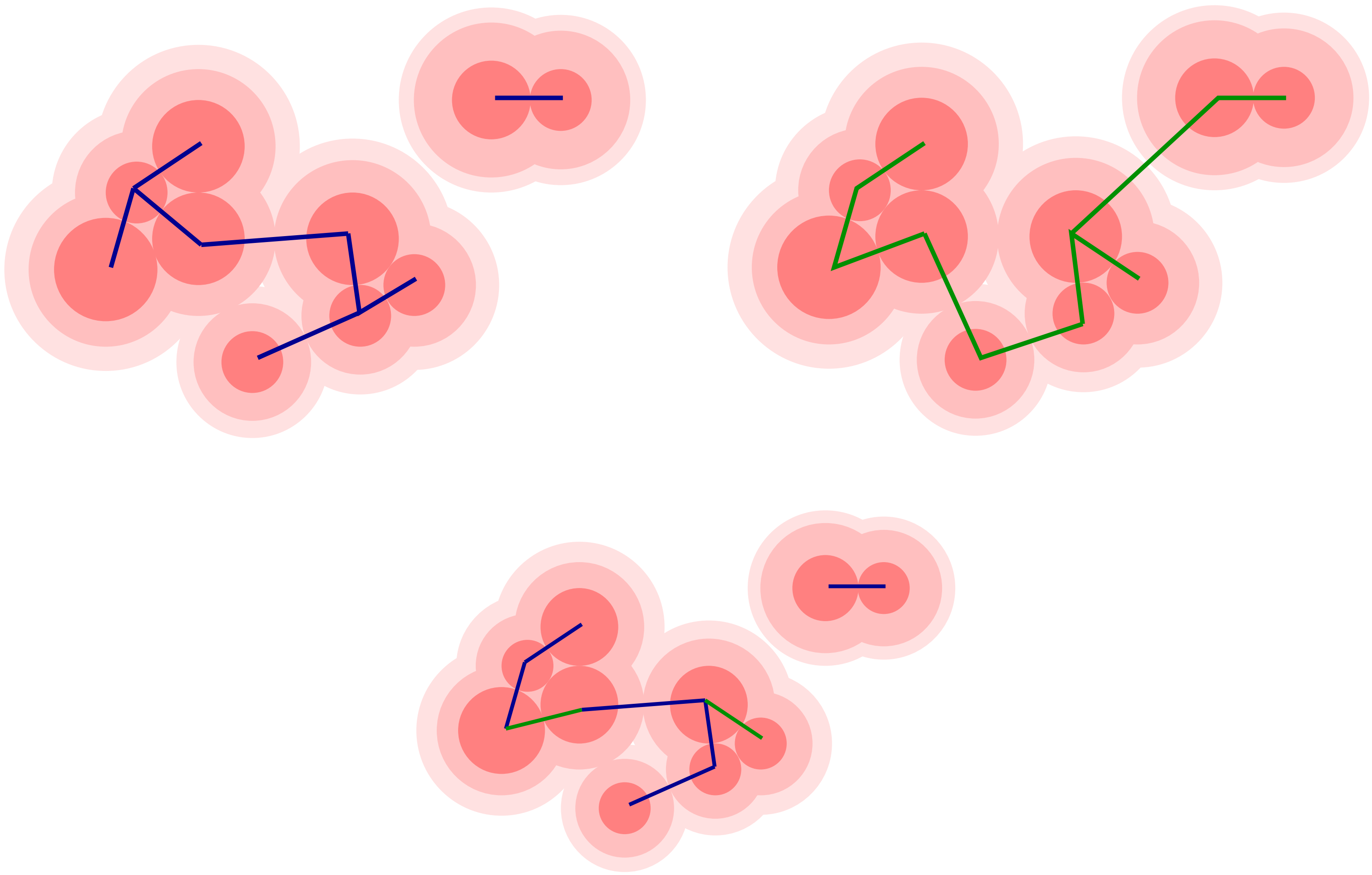
moving from F_{q-} *to* F_{q+}



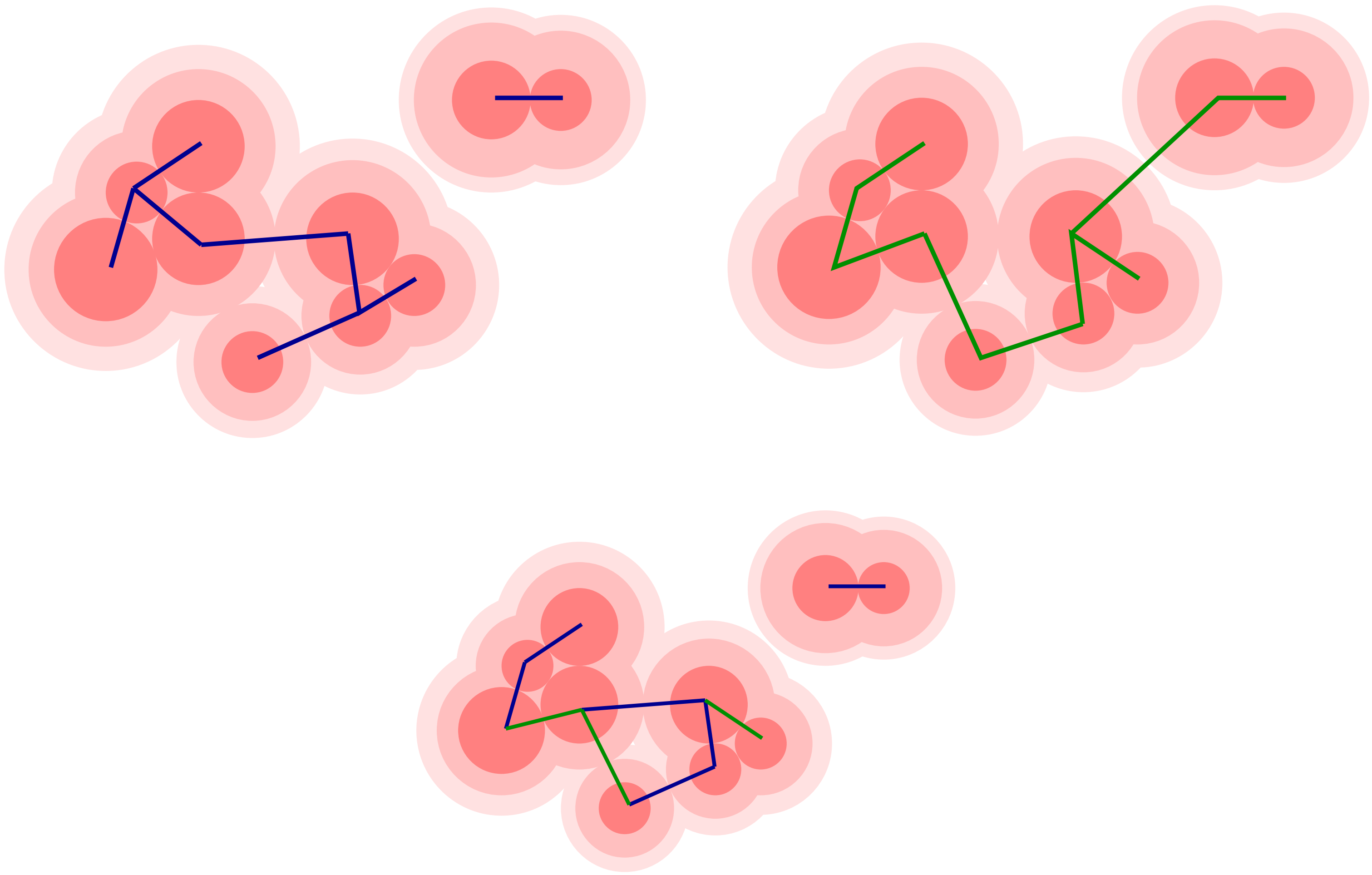
moving from F_{q-} to F_{q+}



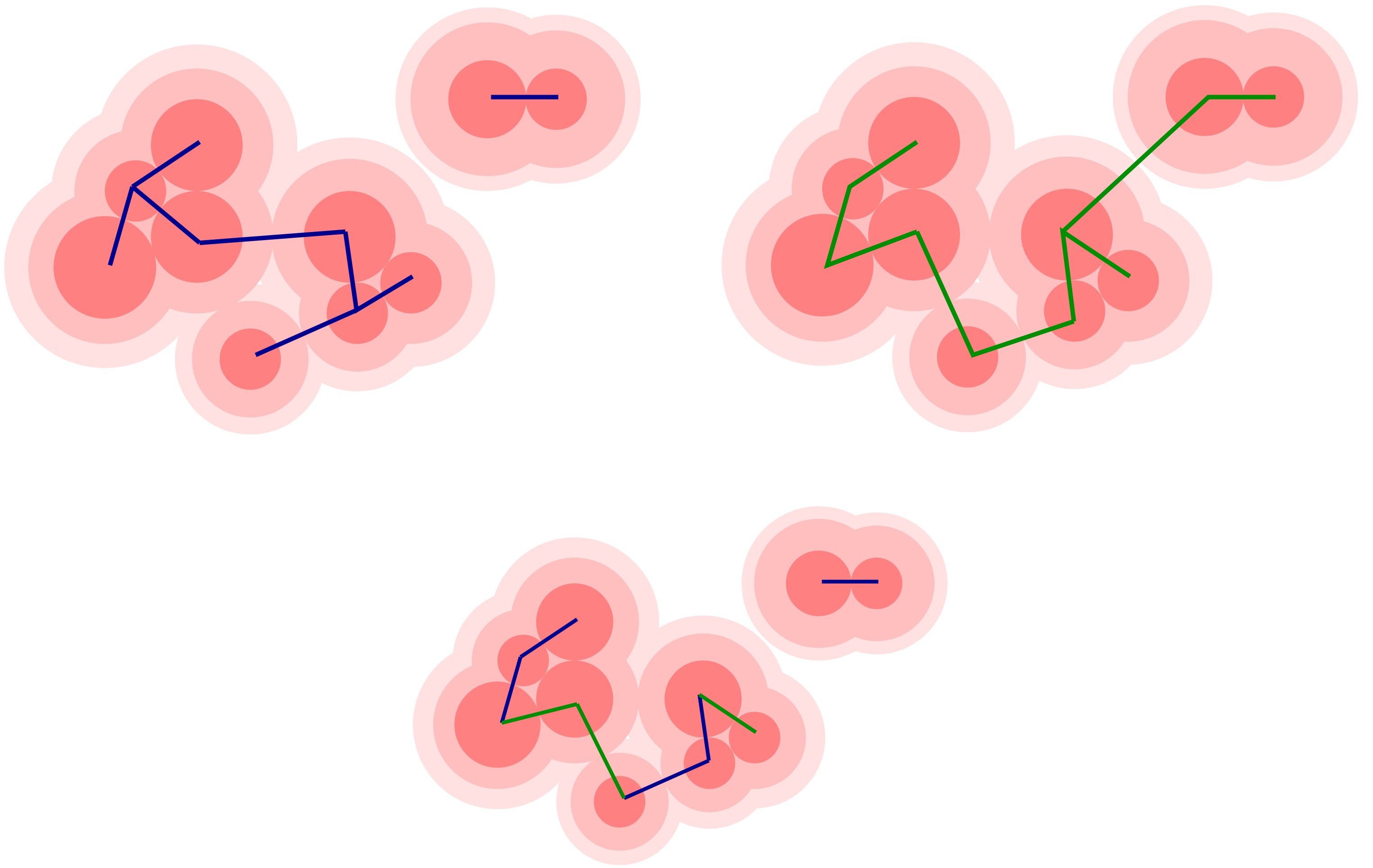
moving from F_{q-} to F_{q+}



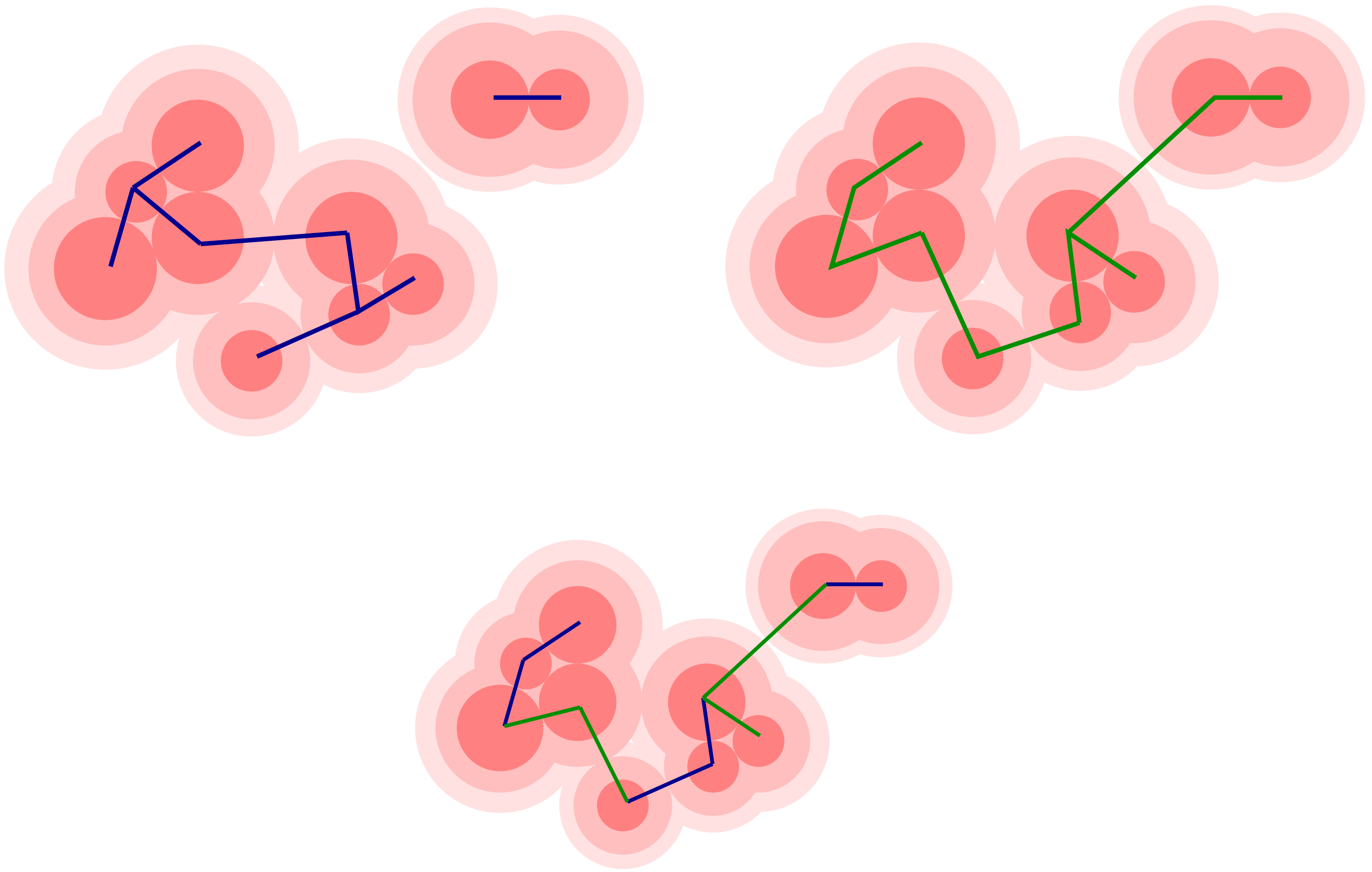
moving from F_{q-} to F_{q+}



moving from F_{q-} to F_{q+}



moving from F_{q-} *to* F_{q+}



From χ_{q-} to χ_{q+}

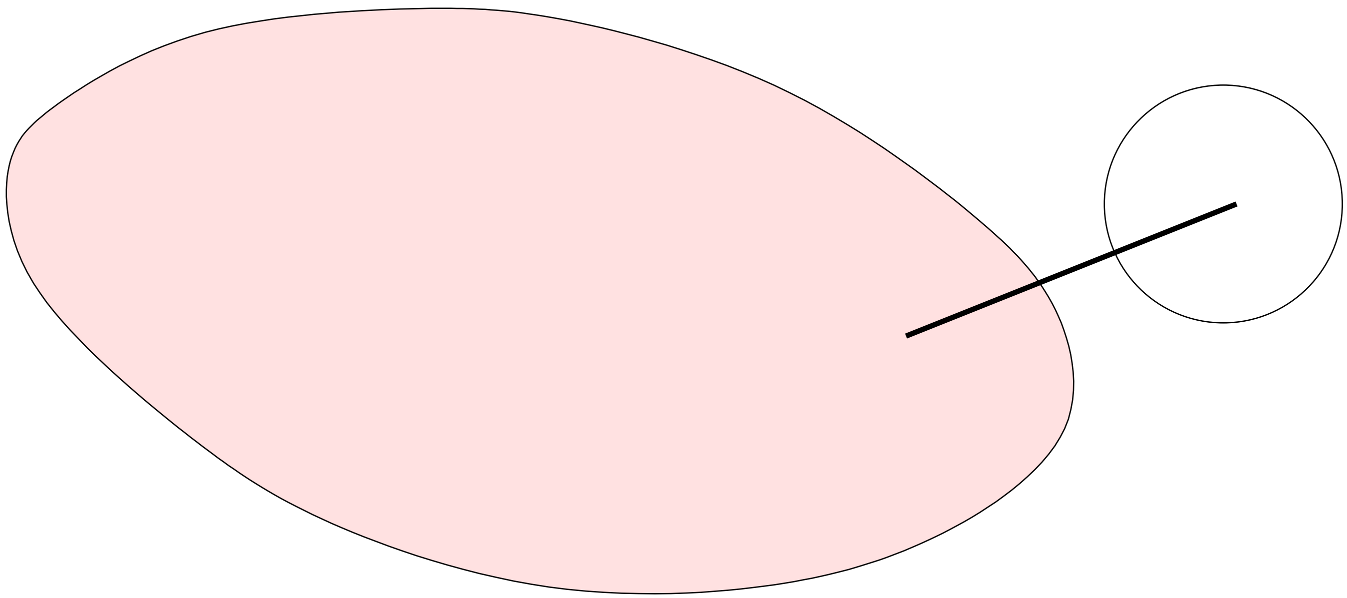
- Through a sequence of steps we have moved from $(\mathcal{S}_q, \chi_{q-}, F_{q-})$ to $(\mathcal{S}_q, \chi_{q-}, F_{q+})$.
- The sets in $\mathcal{S}_q$ which are colored grey in χ_q are colored black in χ_{q-} and white in χ_{q+}.
- We consider these sets one by one and change their color from black to white.
- This sequence of steps takes us from $(\mathcal{S}_q, \chi_{q-}, F_{q+})$ to $(\mathcal{S}_q, \chi_{q+}, F_{q+})$.
- At some step of these two sequences the value of $\alpha()$ should jump from " less than k" to "more than k".
- This is the step of interest.

Recoloring of a set

When the step of interest corresponds to changing the color of some set S from black to white.

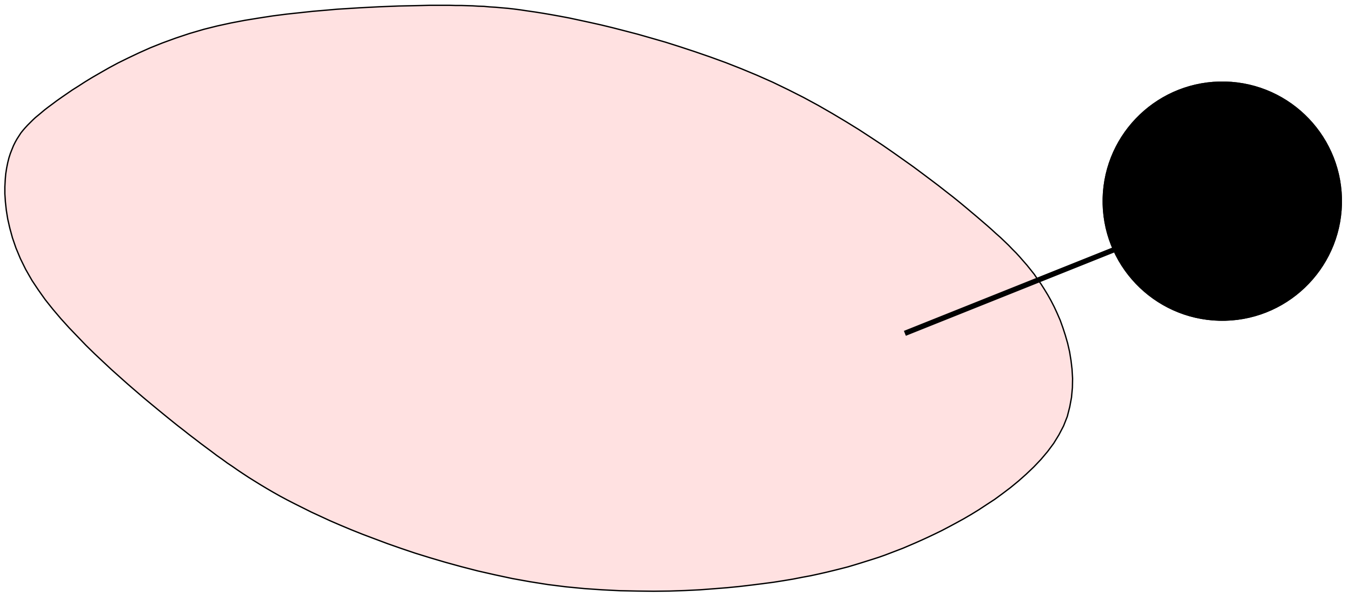
Recoloring of a set

When the step of interest corresponds to changing the color of some set S from black to white.



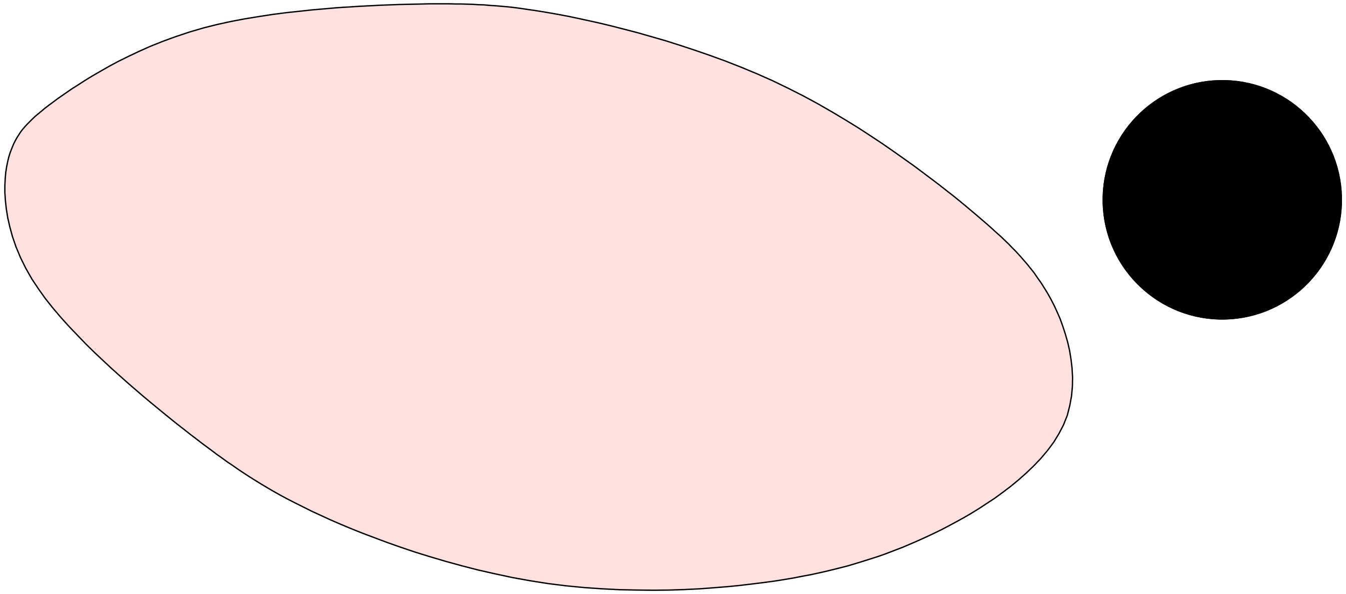
Recoloring of a set

When the step of interest corresponds to changing the color of some set S from black to white.



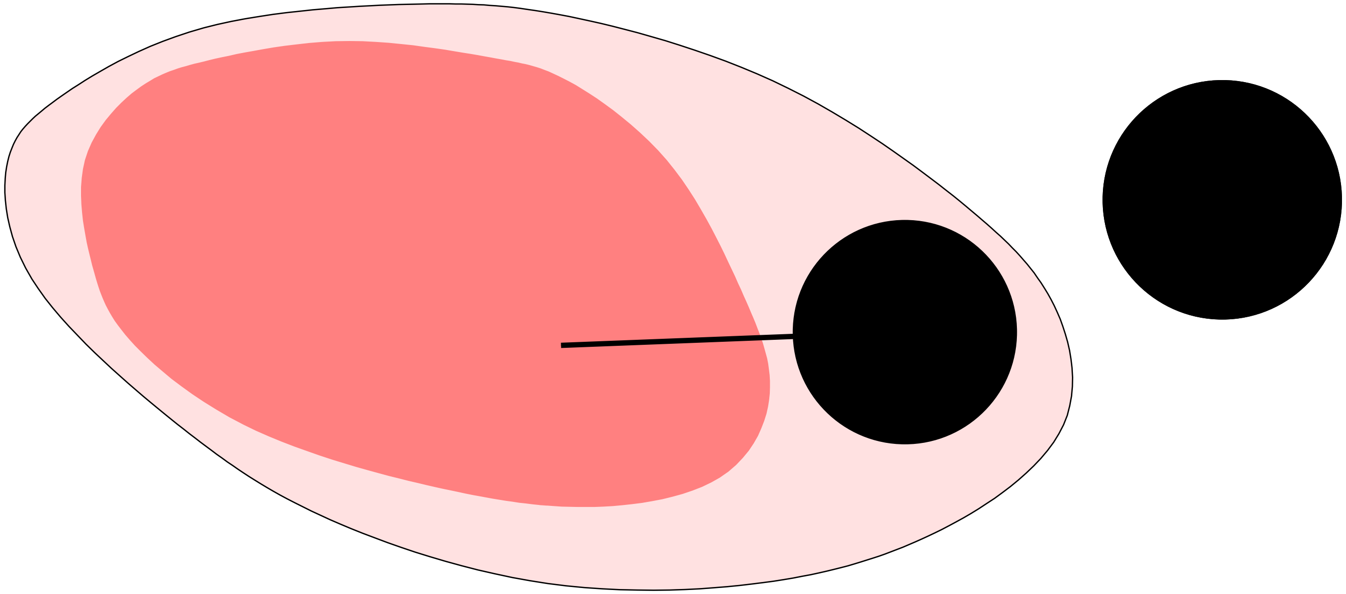
Recoloring of a set

When the step of interest corresponds to changing the color of some set S from black to white.



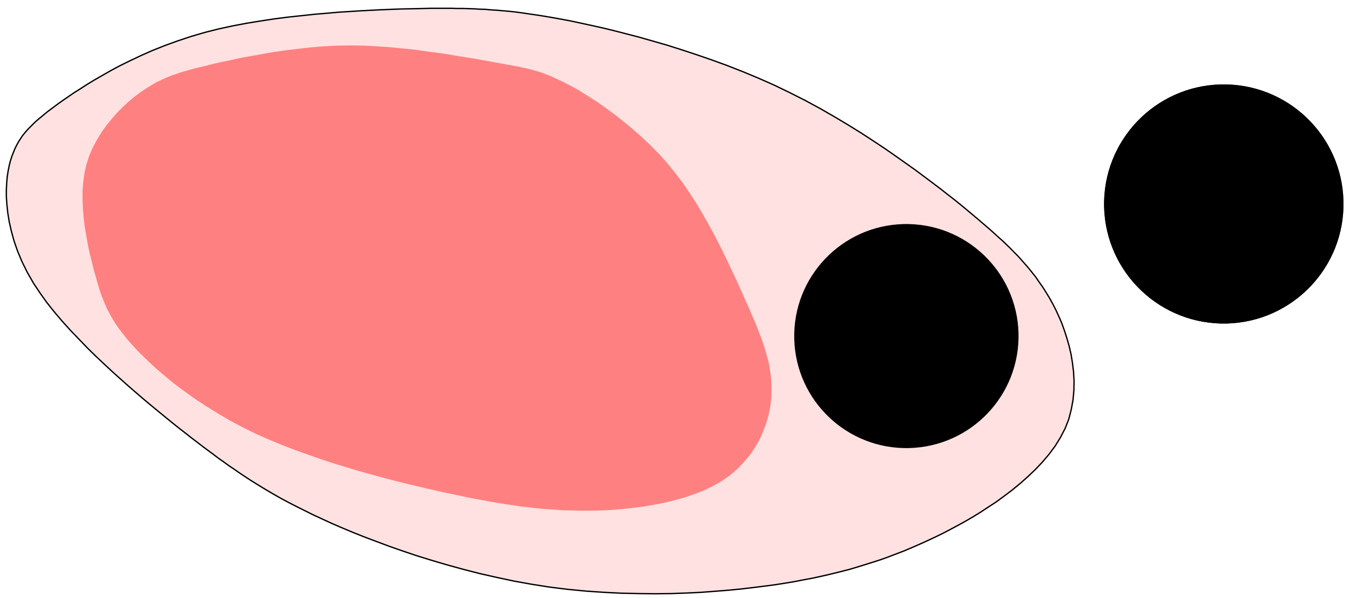
Recoloring of a set

When the step of interest corresponds to changing the color of some set S from black to white.



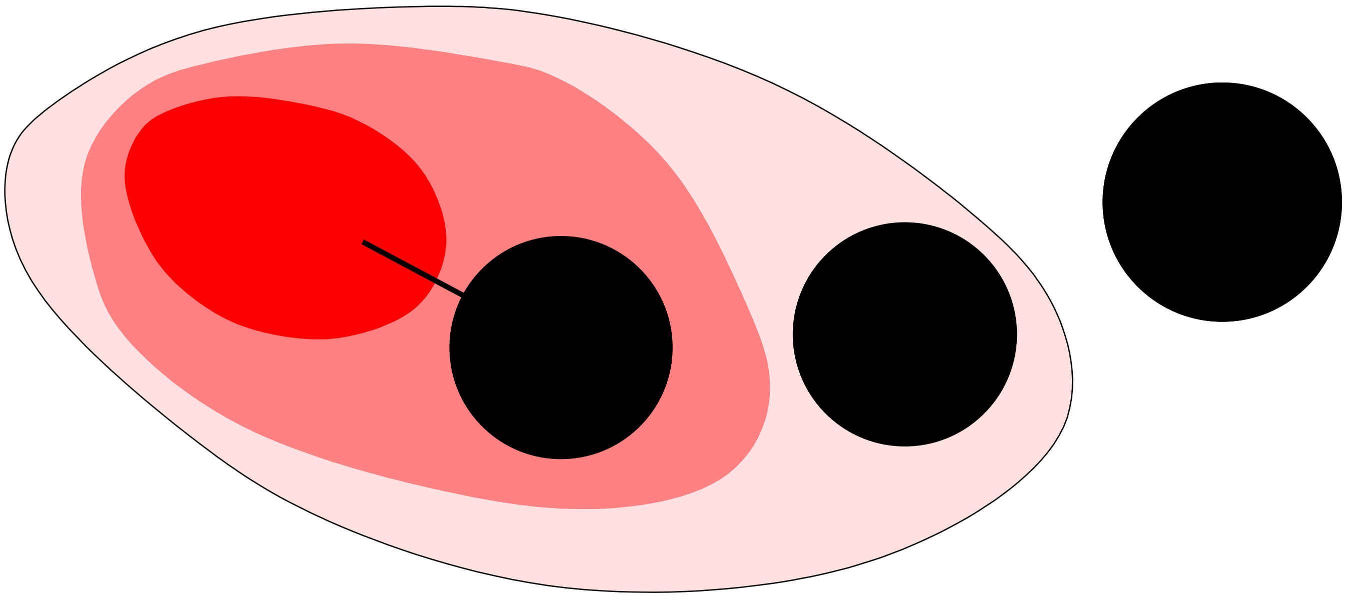
Recoloring of a set

When the step of interest corresponds to changing the color of some set S from black to white.



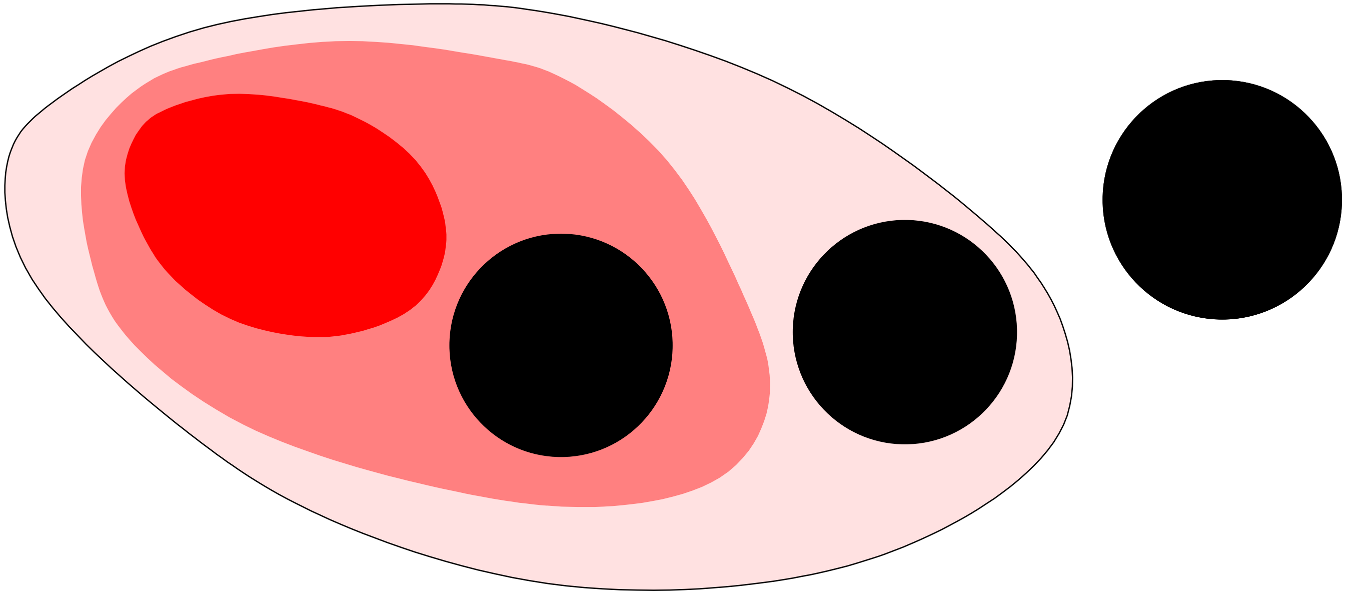
Recoloring of a set

When the step of interest corresponds to changing the color of some set S from black to white.



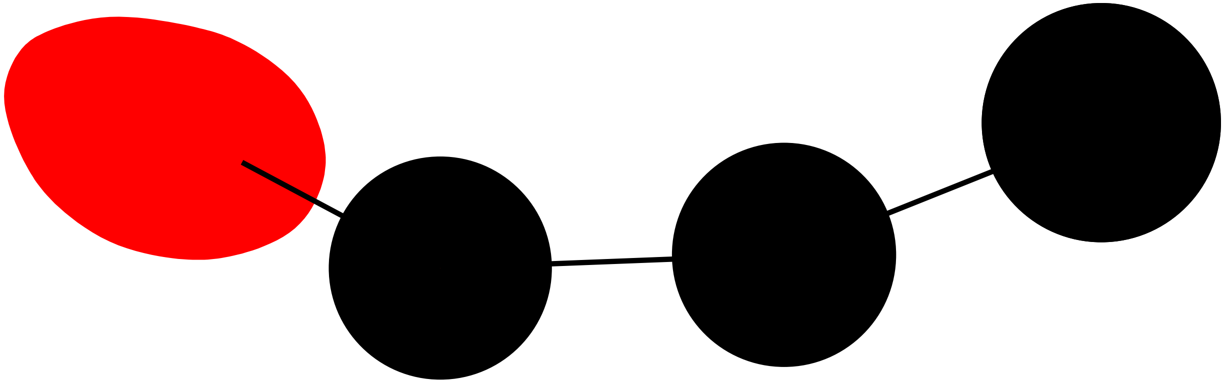
Recoloring of a set

When the step of interest corresponds to changing the color of some set S from black to white.



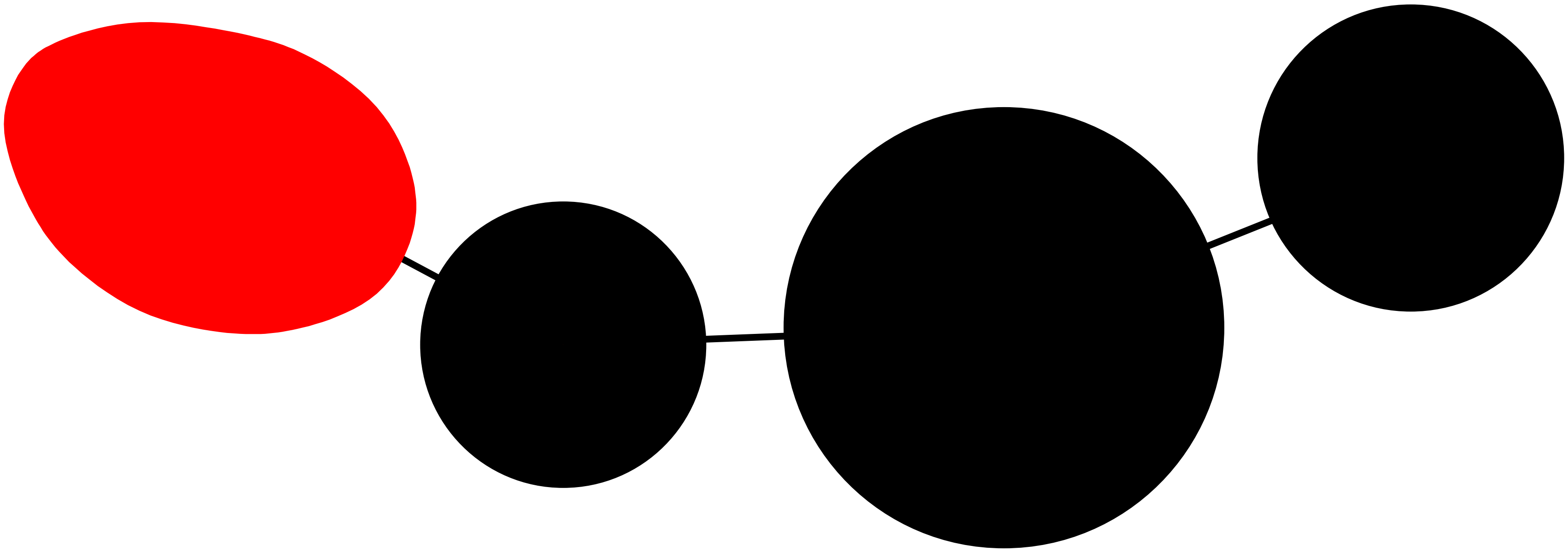
Recoloring of a set

When the step of interest corresponds to changing the color of some set S from black to white.



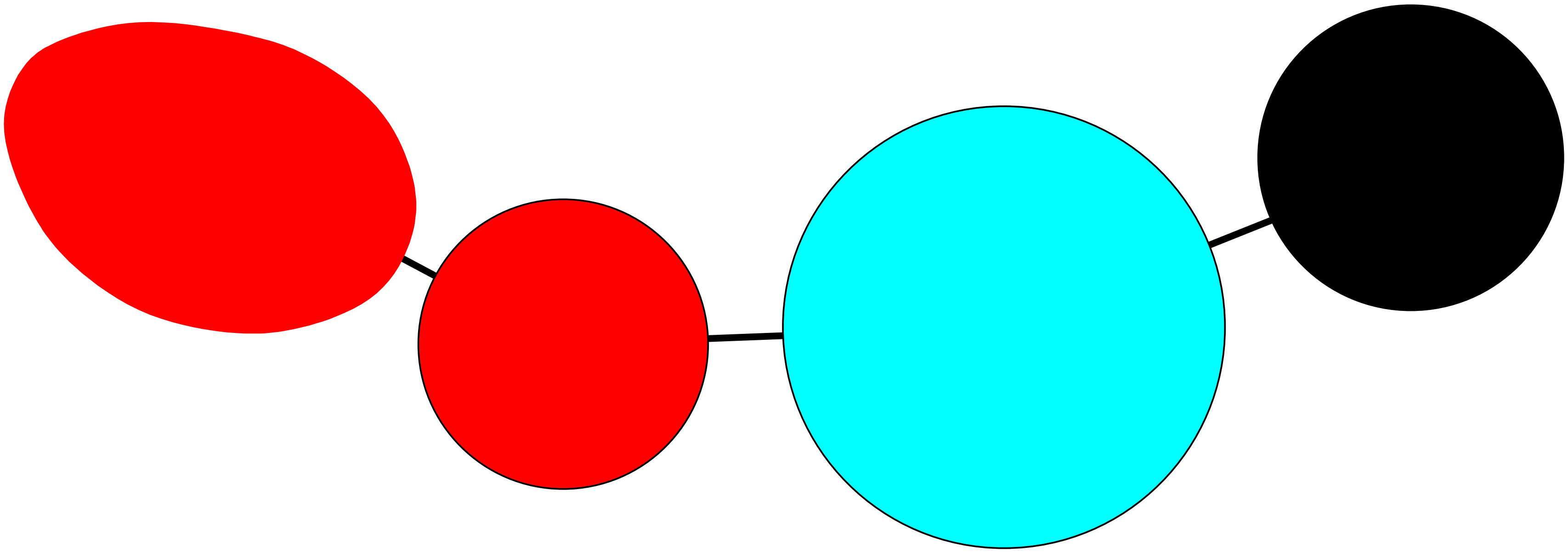
Note that the number of red vertices is less than k while the red+black exceeds k.

Picking k vertices from a chain



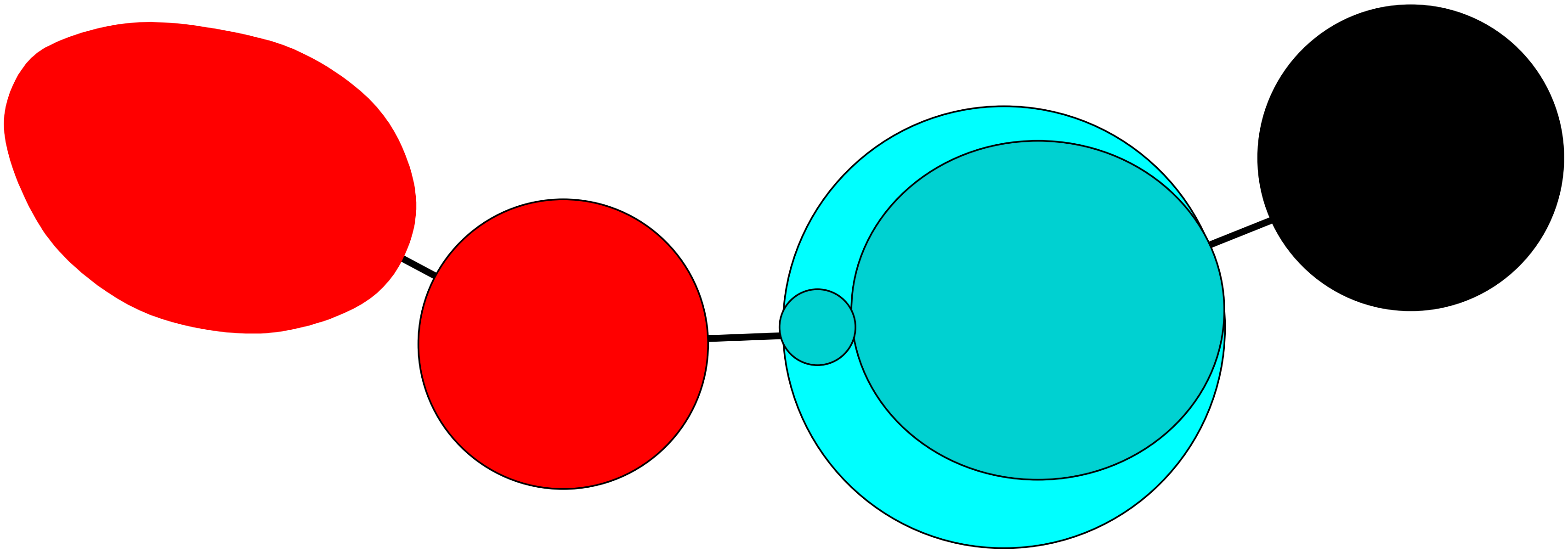
We will pick k contiguous vertices. All the vertices in the red blob and some in the black blobs would be included.

Picking k vertices from a chain



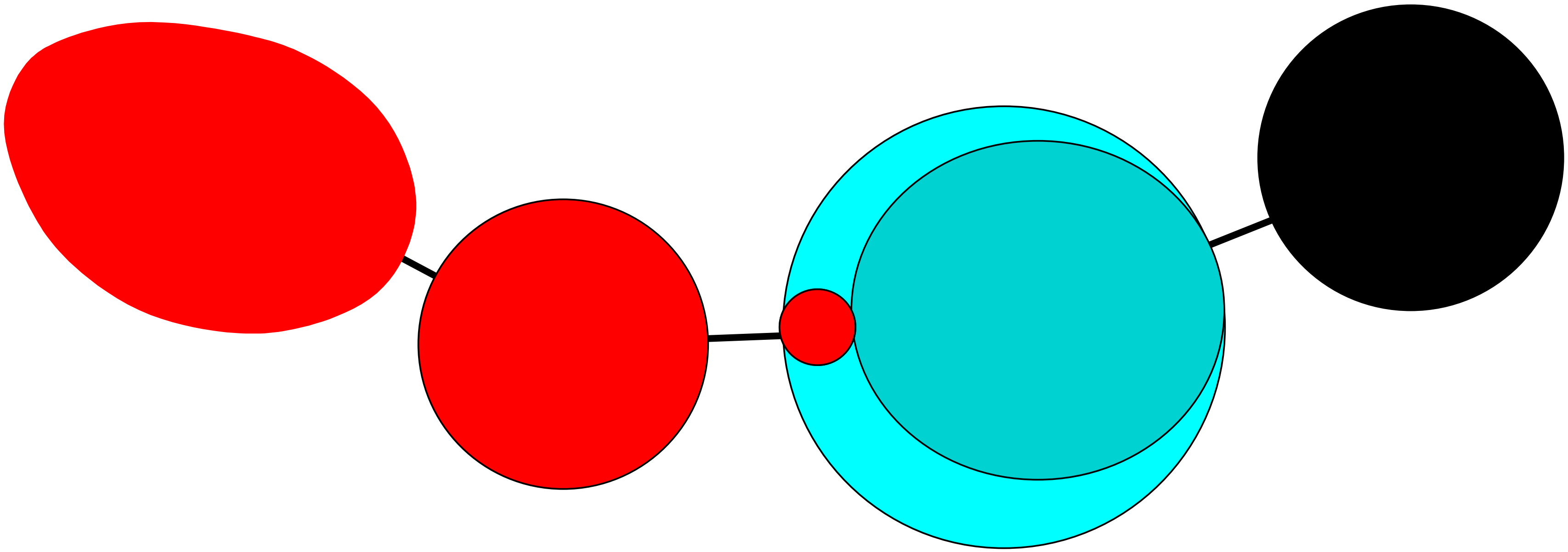
We traverse the chain (starting from the red blob), including vertices, till we reach a black blob, whose inclusion would take the number of vertices beyond k.

Picking k vertices from a chain



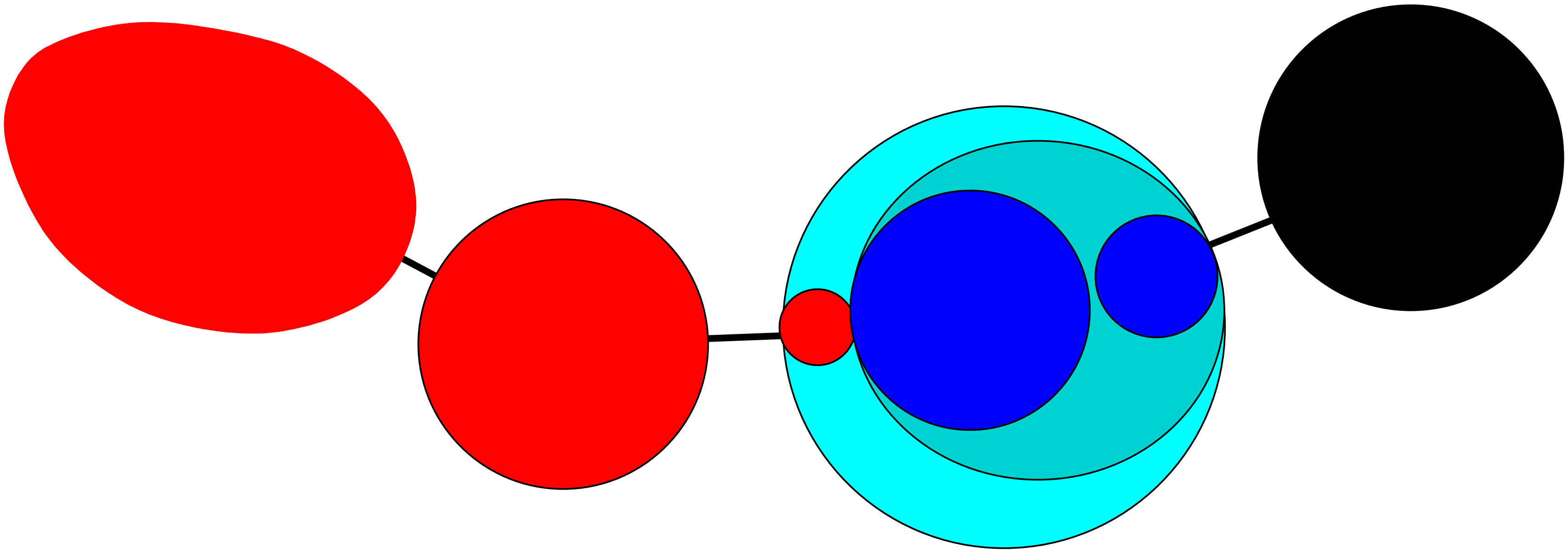
We now split this blob by looking at the two sets which merged to form it.

Picking k vertices from a chain



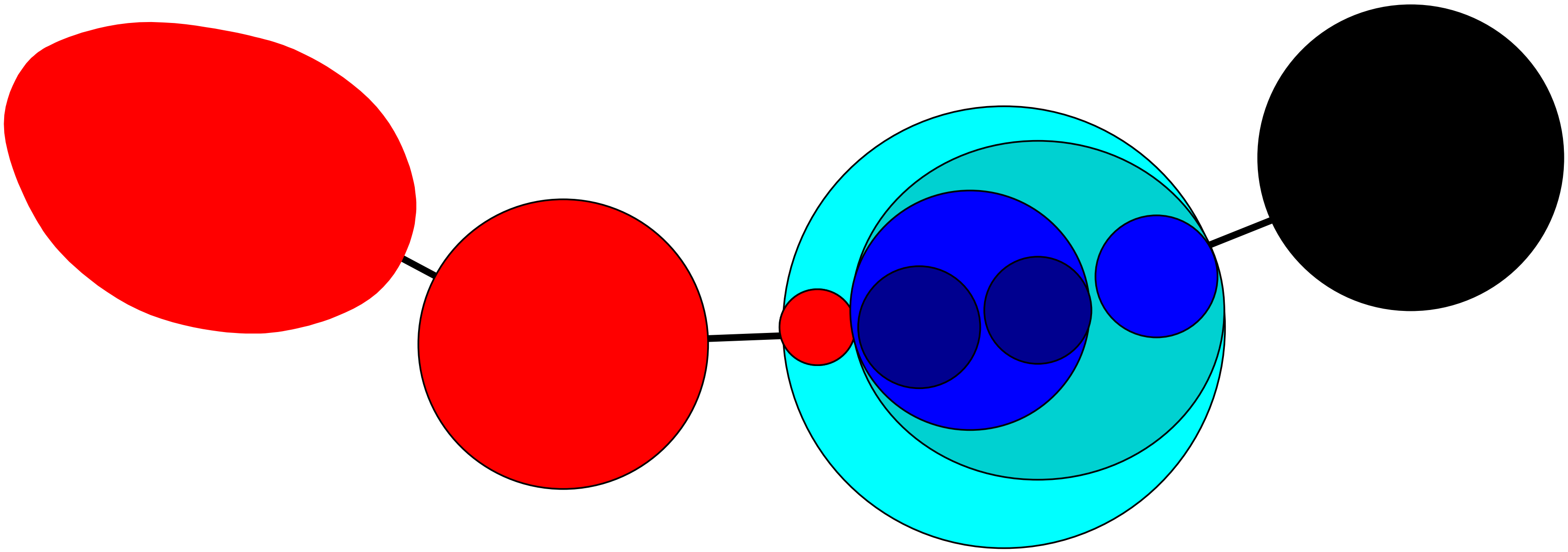
If including the first of these keeps us under k we pick these vertices.

Picking k vertices from a chain



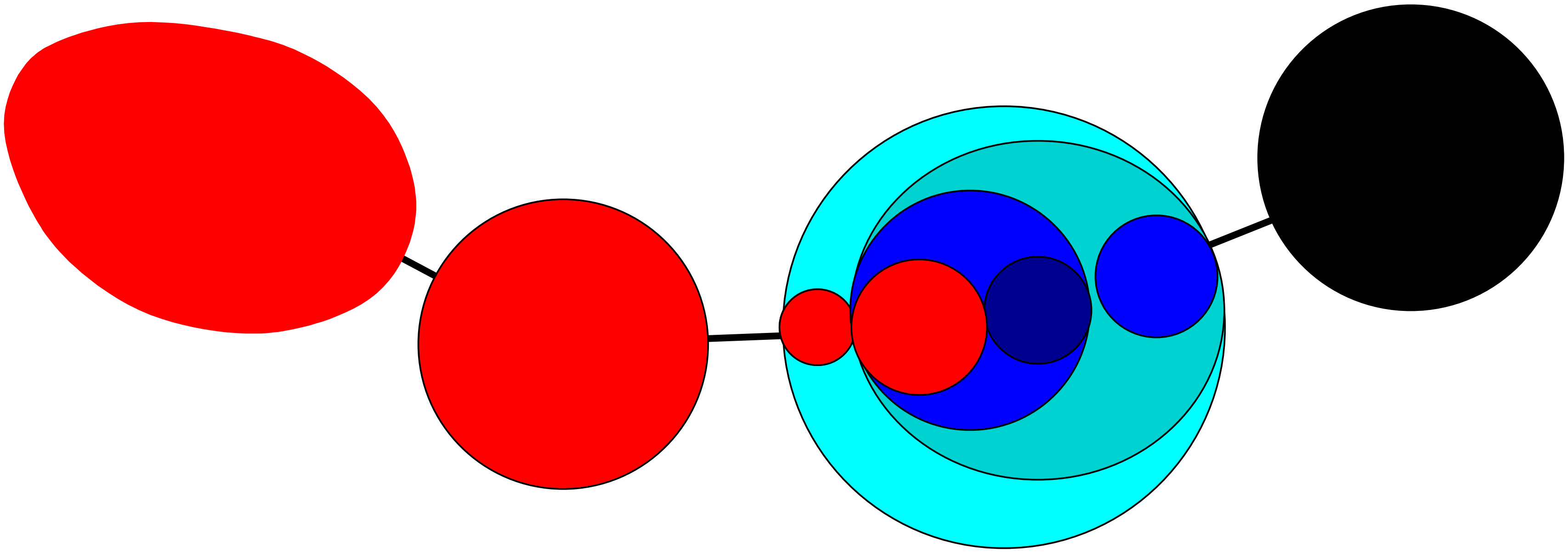
Since including the second takes us beyond k we split it further.

Picking k vertices from a chain



If including the first still takes us beyond k we split it.

Picking k vertices from a chain



Finally, including the first set takes us to exactly k vertices. We pick these and stop.

The case of swapping edges

- We now consider the setting when including an edge $e+$ and removing edge $e-$ causes one of the trees in the forest to have more than k vertices.
- If we include both edges $e-$ and $e+$ we would form a cycle.
- Removing the edge $e+$ from this cycle would cause some more vertices to be pruned and this would bring the number of vertices in the tree to below k.

Illustration of edge swapping

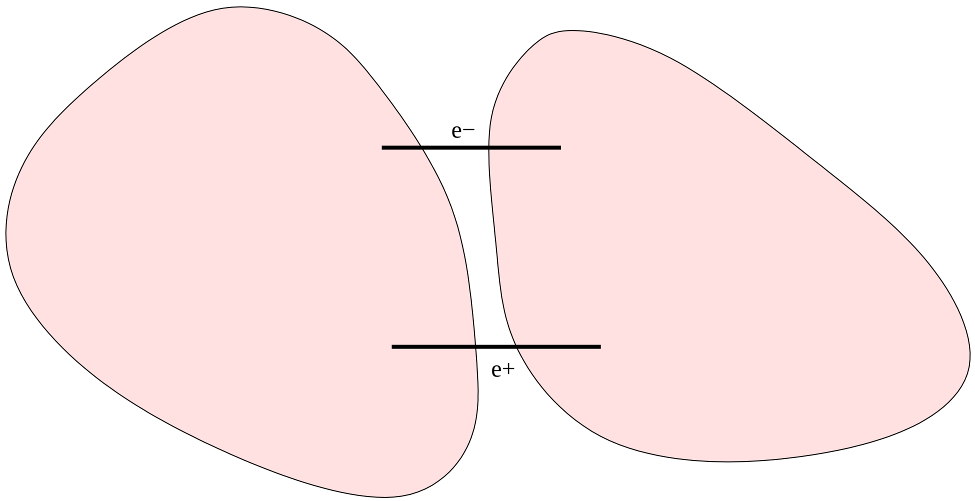
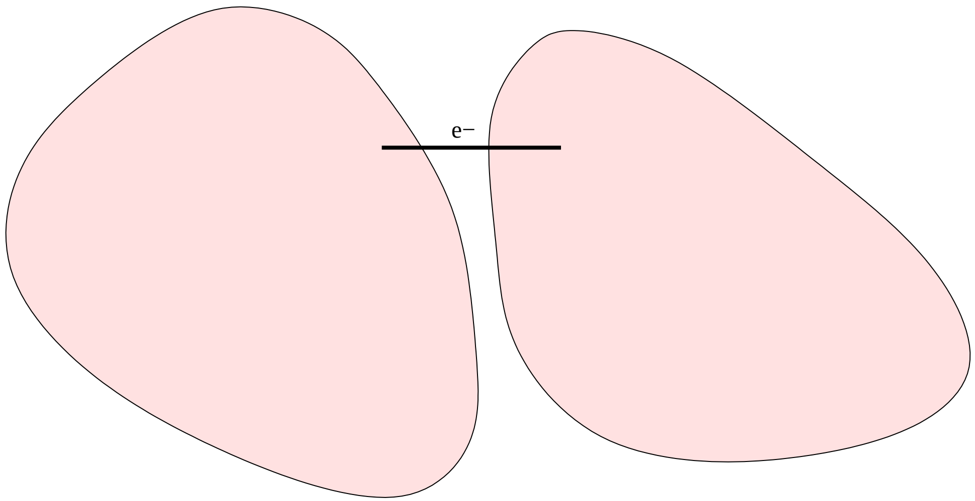
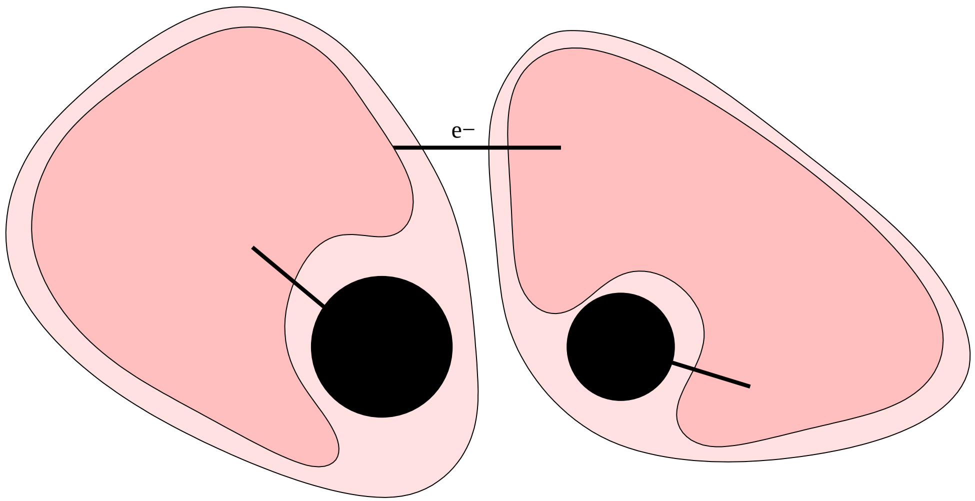


Illustration of edge swapping



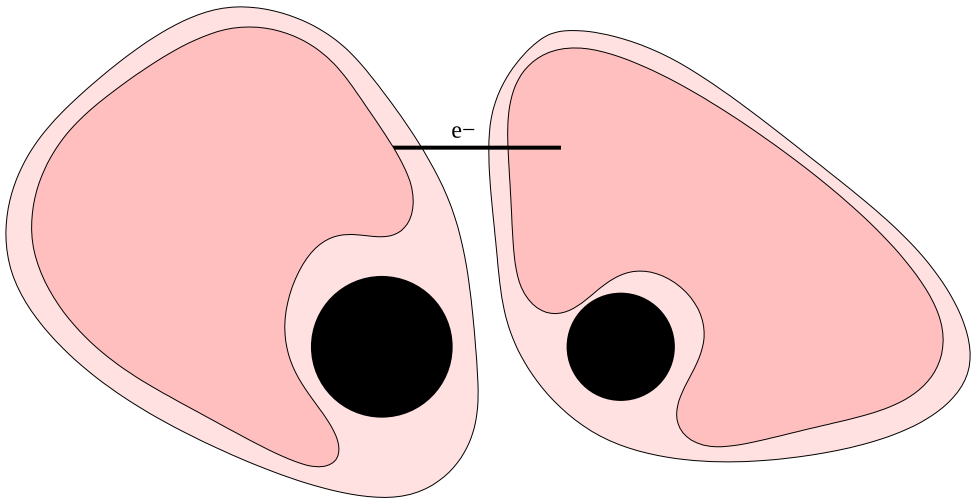
We remove edge $e+$.

Illustration of edge swapping



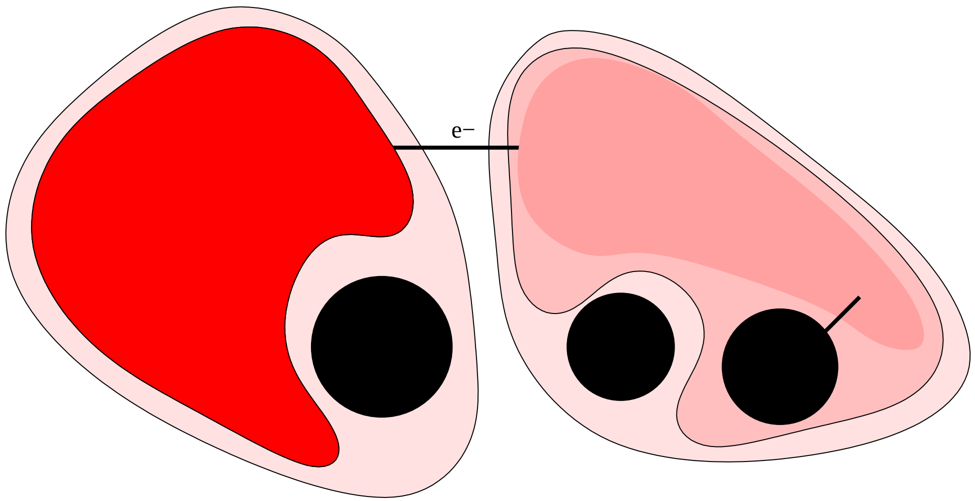
Find maximal black sets which have only one edge incident to them.

Illustration of edge swapping



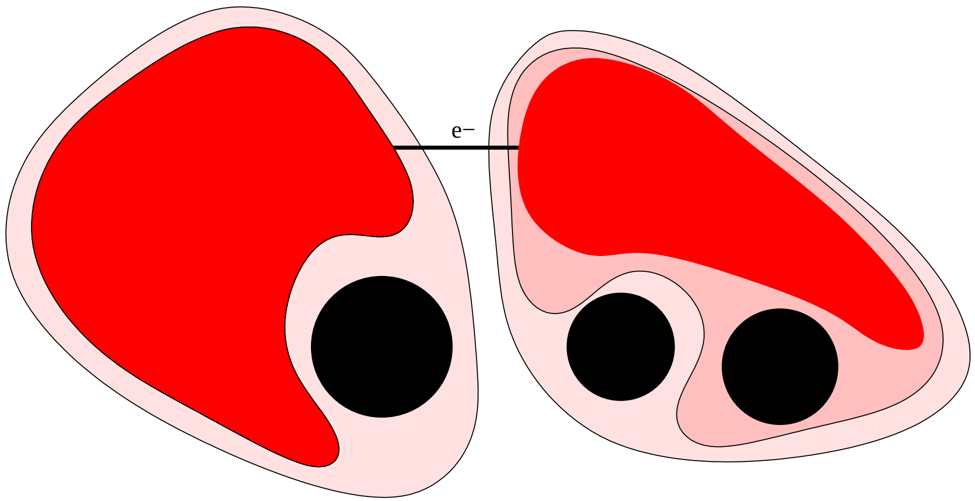
Prune the edges corresponding to black leaves.

Illustration of edge swapping



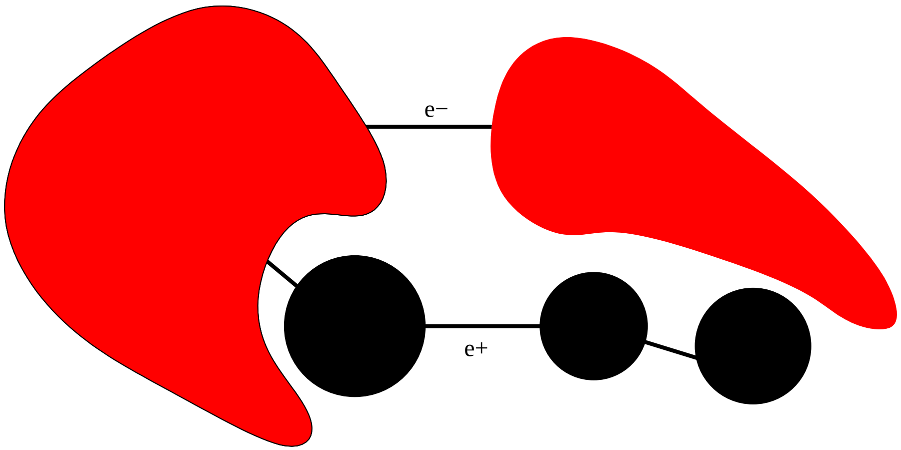
Again find maximal black sets which have only one edge incident to them. No such set exists in the left blob and so the remaining vertices are colored red.

Illustration of edge swapping



Prune edge. Now there is no maximal black set which has only one edge incident to it and so remaining vertices are colored red.

Illustration of edge swapping



The red vertices are contiguous and less than k in number. There is a chain of black vertices. Since red+black is more than k, using the same procedure as before we can pick k vertices.

The Analysis

Our analysis of the approximation ratio is similar to the Goemans Williamson analysis.

Let T be the tree picked. We argue that in each iteration the total degree (in T) of the active sets is at most twice the number of such active sets all of whose vertices are picked.

Interestingly, in an iteration the edges of T might form a cycle on the active sets. However, there is only one such cycle and so the GW analysis continues to work.

The analysis shows that the cost of T is at most $2(qk - \pi(Q))$ where Q is the minimal set in $\mathcal{S}_q$ containing all vertices of T.

We wanted to show that T has cost at most $2(qk - \pi(Q))$ where Q is any set in $\mathcal{S}_q$ containing all vertices of T. This requires one more idea.

Conclusions

- The running time of the procedure is $O(mn^4 \log n)$. There is much room for improvement here.
- The equivalence of unrooted and rooted versions of k-mst implies a 2-approximation for the rooted version as well.
- As observed by Johnson et.al., a 2-approximation for k-mst also yields a 3-approximation for the unrooted version of the budget k-mst problem.
- It should be possible to improve the approximation guarantee for budget k-mst and point-to-point orienteering using some of these ideas.