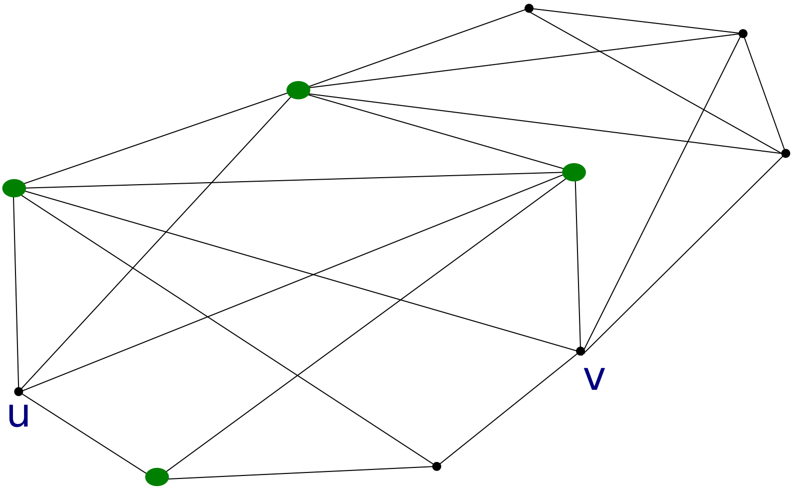
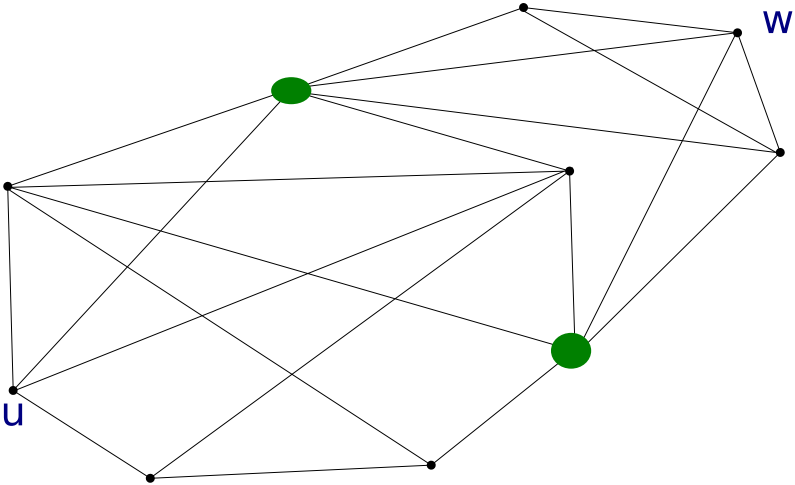
What is a k-cut?

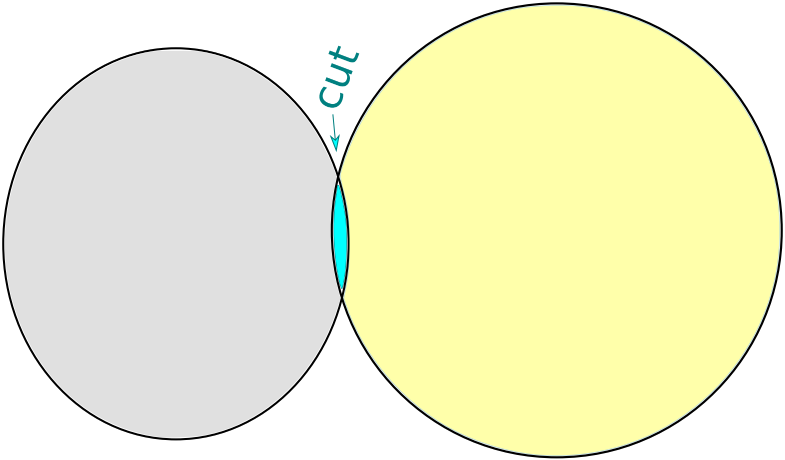


Vertices u and v are 4-connected. Higher? No; there is a 4-cut.

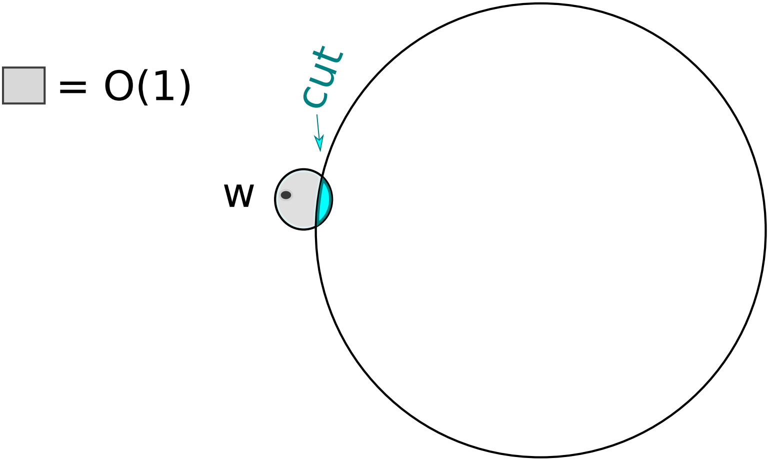
What is a k-cut?



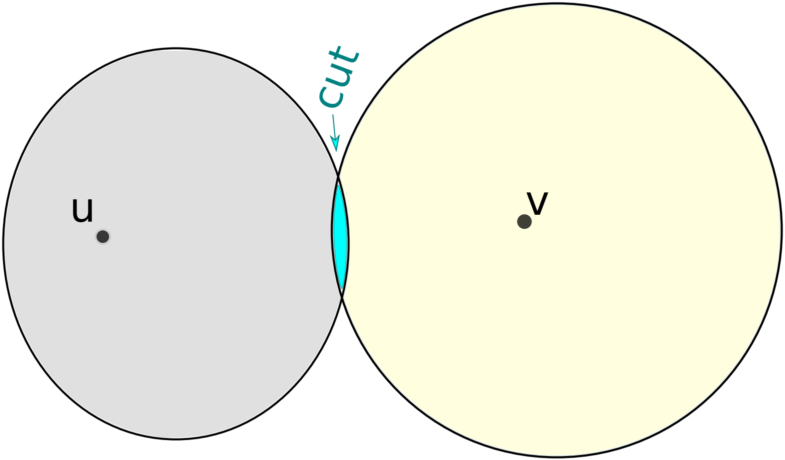
How to find a cut



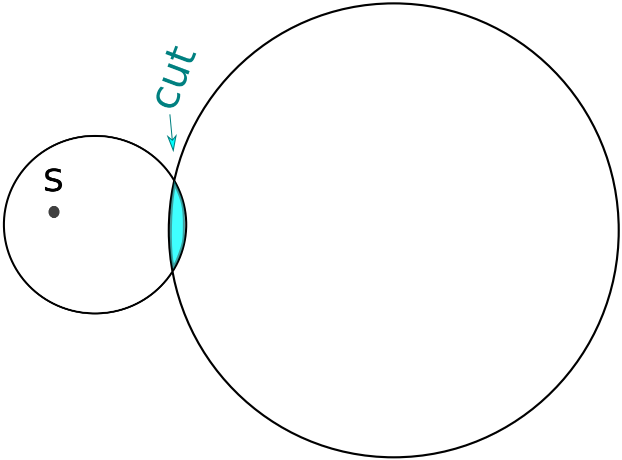
How to find a cut



How to find a cut



How to find a cut



How to find a cut

Assume there is a cut.

- If the cut is quite balanced, we are quite likely to find it.
- If the cut has a tiny side, we are quite likely to find it.
- For any amount of “balancedness” in between, we are quite likely to find it.

So we can run different approaches that are tailored to different values of “balancedness”.
As soon as one of them successfully finds a cut, we are done.
If there is a cut, we are not very likely to have overlooked it.
So if none of them finds a cut we return that none exists.

Weird stuff?

- $\tilde{O}(f(x))$, aka. "O-tilde notation". Hides $log(f(x))$-factors.

  Ex: $\tilde{O}(n^c)$ could be n^c, or $n^c \log n$, or $n^c(\log n)^{1001}$

- w.h.p, aka. "with high probability".

  Means that: the bigger the input to the algorithm gets,

  the closer to 1 is the probability that it works.

  That is, the probability converges to one, as the input size goes to infinity: $p \to 1$ as $n \to \infty$.