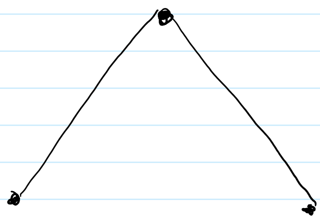
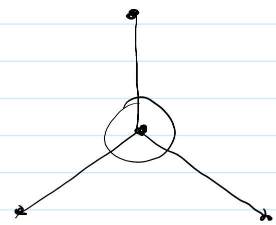
- most of the problems we face in practice are NP-hard.
  - MST vs. steiner tree



MST



Steiner tree

  - Shortest path vs. Travelling Salesman (TSP)
    (visit all vertices in the graph
    find the shortest cycle)

- NP-hardness
  - It is hard to solve ALL problems of this type.
  - It may not be hard to solve any specific problem of this type.

- Clever search algorithms
  - search algorithm
    - 3-SAT n variables
    - enumerate 2^n possible assignments for these n variables
    - not feasible when $n \geq 50$.
  - $x_1 \vee x_2 \vee x_3$

    $x_1 = 0$, $x_2 = 0$, $x_3 = 0$
  - popular heuristics for 3-SAT: DPLL

    ① If a clause has only 1 literal, then it must be true

    x_1 $\quad \bar{x}_2$

    modify clauses when fixing value of variable

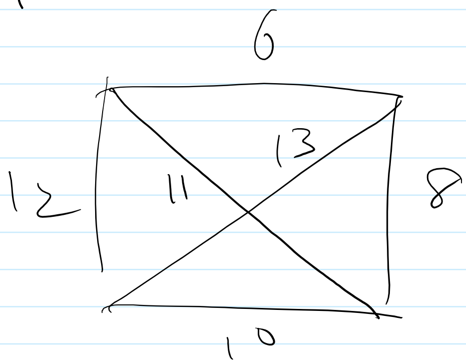
    $x_1 \vee x_2 \vee x_3$ → $x_1 = 0$ → $x_2 \vee x_3$

    $x_1 \vee x_2 \vee x_3$ → $x_1 = 1$

$x_1 \vee x_2 \vee x_3$ — $x_1=0$ / $x_1=1$ → delete the clause

② If a variable only appears in 1 form (x_i or $\bar{x}_i$), set it to the correct value.

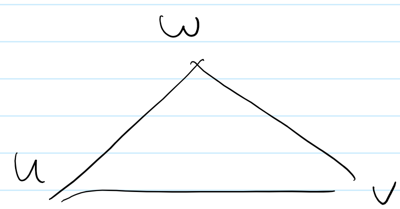
- rely on approximation algorithms.
  - Find approximate solutions instead of optimal solutions.
- e.g. TSP



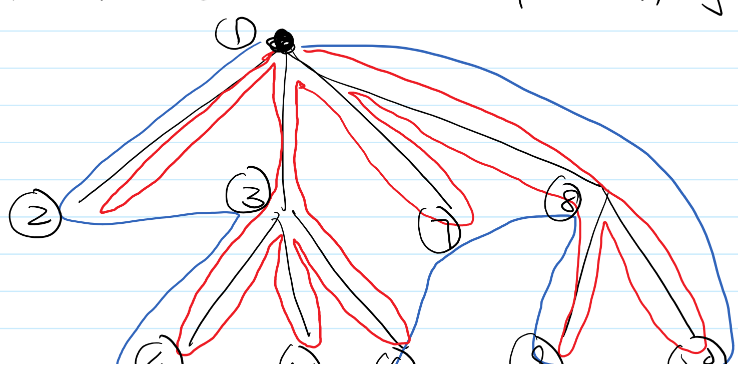
- Say an algorithm is a ρ-approximation, if it can always output a solution whose cost is no more than $\rho \cdot OPT$
  (OPT: cost of optimal solution)

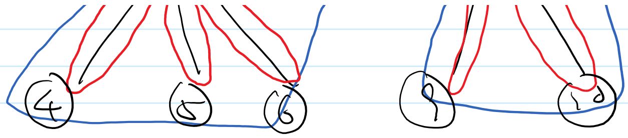
- simple approximation algorithm for TSP (with triangle inequality)

  $W_{u,v} \le W_{u,w} + W_{w,v}$



- ① Find a MST
  ② visit vertices in DFS order (preorder) of the MST.





- Claim: This is a 2-approximation algorithm.
- Proof: cost of MST ≤ cost of TSP

  cost of alg ≤ 2X cost of MST

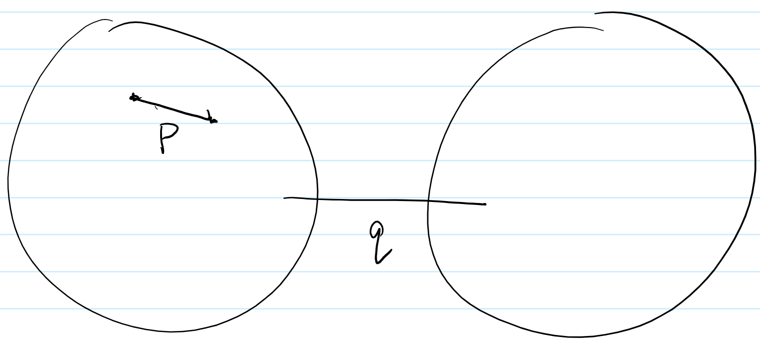
  cost of blue cycle   cost of red cycle

- FPTAS (fully polynomial time approximation scheme)

  there is an algorithm that can find a $(1+\varepsilon)$ approximate solution in $\text{poly}(n, \frac{1}{\varepsilon})$ time.

  - Knapsack

- average case analysis
  - input is generated by nature, not by an adversary.
  - input is generated by some distribution, the algorithm should work with high probability under this distribution.
  - average case ≠ randomized algorithm.
  - example: community problem (simple ver)



people in same community connected w.p. p

——— different ——————— w.p. q

$p > q$

BALANCED-SEPARATOR: find a cut that divide people into two equal-size groups with fewer edges.

↑

equal-size groups with fewer edges.

↑ NP-hard.

- For this particular distribution. communities can be found by applying "spectral clustering" (PCA on the adjacency matrix)

- use hardness for crypto