Advanced Algorithms

Lecture 10: MST (contd.), local search

Announcements

- HW 2 due tomorrow!
- HW 1 grading, comments (Vivek Gupta)

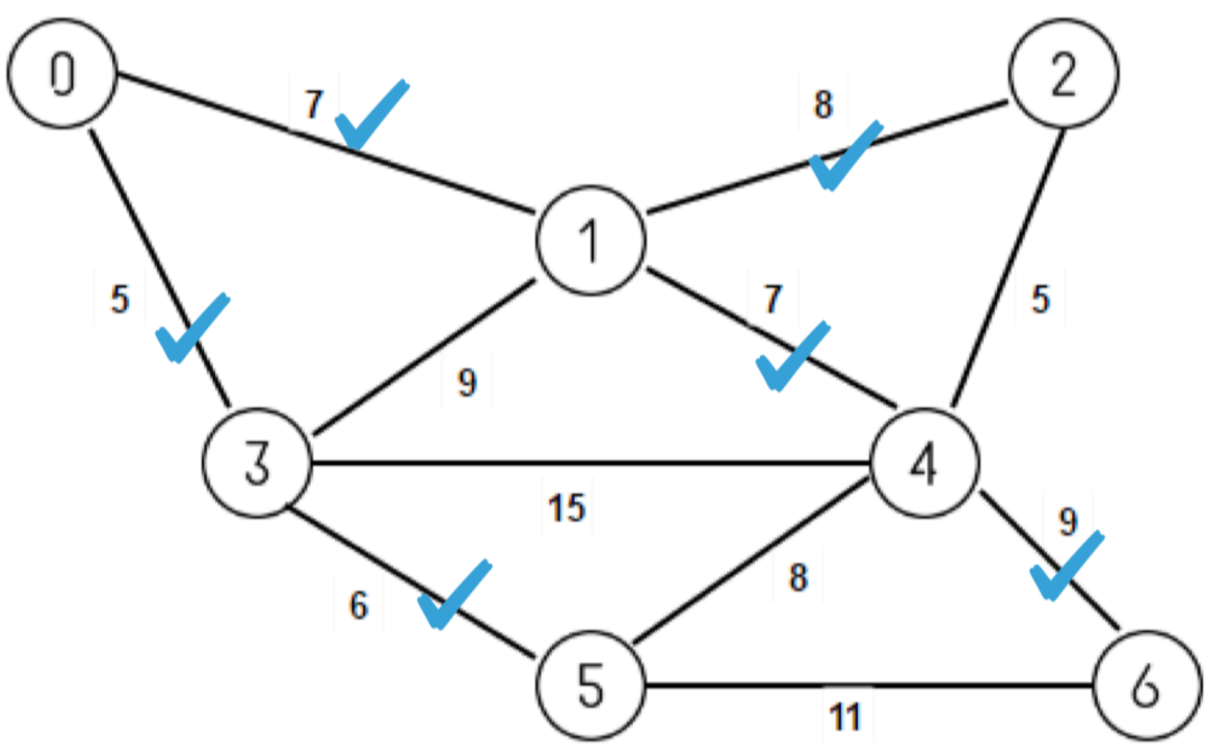
Greedy algorithms – comments

- Usually “easy” to come up with (we are naturally myopic)
- Usually not optimal — examples, Traveling salesman, set cover, …
- (Due to this..) analysis is usually tricky

Example 2: spanning trees

Problem: let G = (V, E) be a (simple, undirected) graph with edge weights $\{w_e\}$ (>0). Pick a subset of the edges, such that (a) all vertices are "connected", (b) total weight of edges is minimized

(Communication backbone in a network)



Greedy strategy

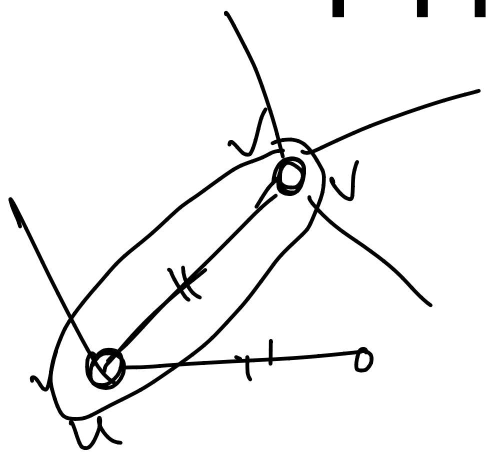
- **Goal:** need to connect all vertices to one another

- Prim: Start with one vertex, add a new vertex to connected set each time

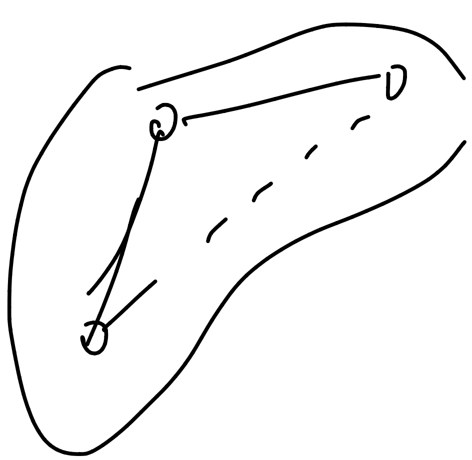
- Kruskal: Add edges one at time, choose min weight edge that isn't "redundant"

Surprise: both turn out to be optimal!

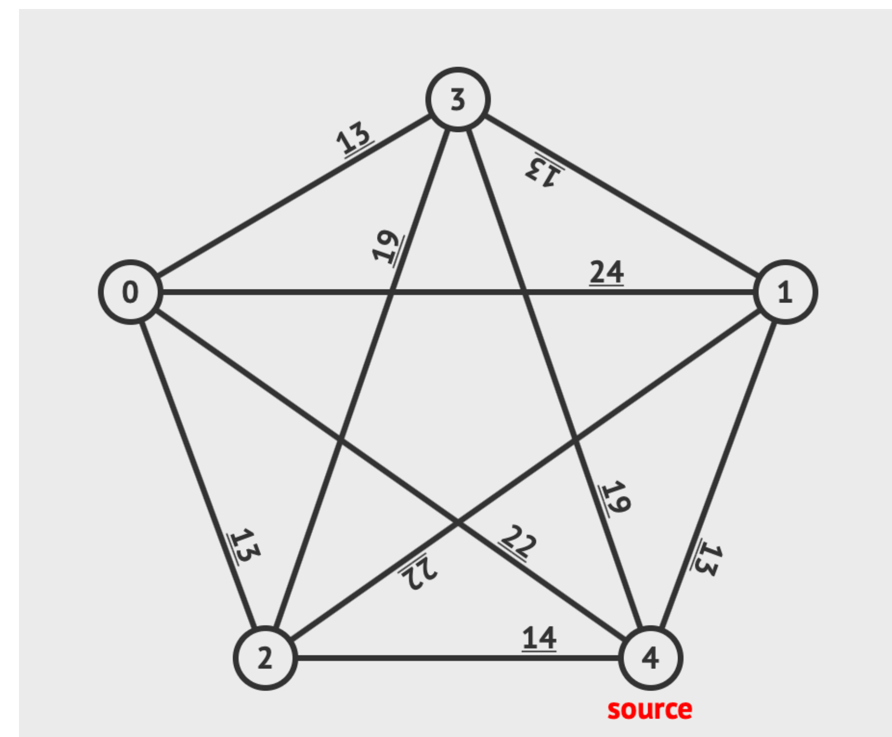
Prim’s algorithm



- start with $S_1 = \{u\}$
- for t = 1, … , n-1:
  - add least wt edge out of S_t



https://visualgo.net/en/mst



Correctness

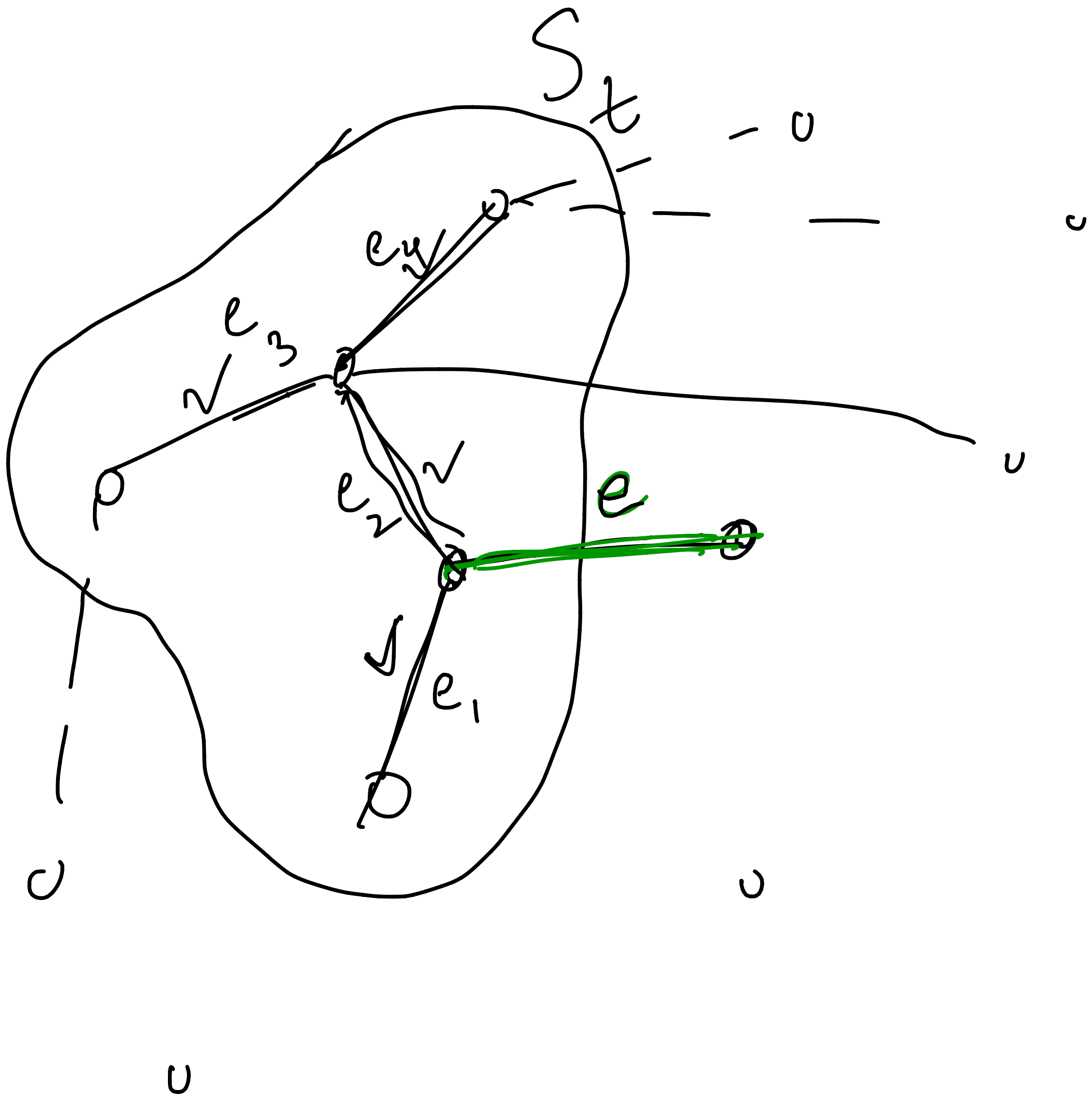
- **Observation:** at each iteration, we have a *set of connected vertices* — S_t

- Will show: There exists a min spanning tree for the *full graph* that contains all edges chosen so far — **structural assumption**

Inductive proof: assuming there's an MST for the full graph containing edges added until t, prove that there's an MST for the full graph containing edge added at $t+1$

$t = n-1$.

Proof of "opt prefix" property

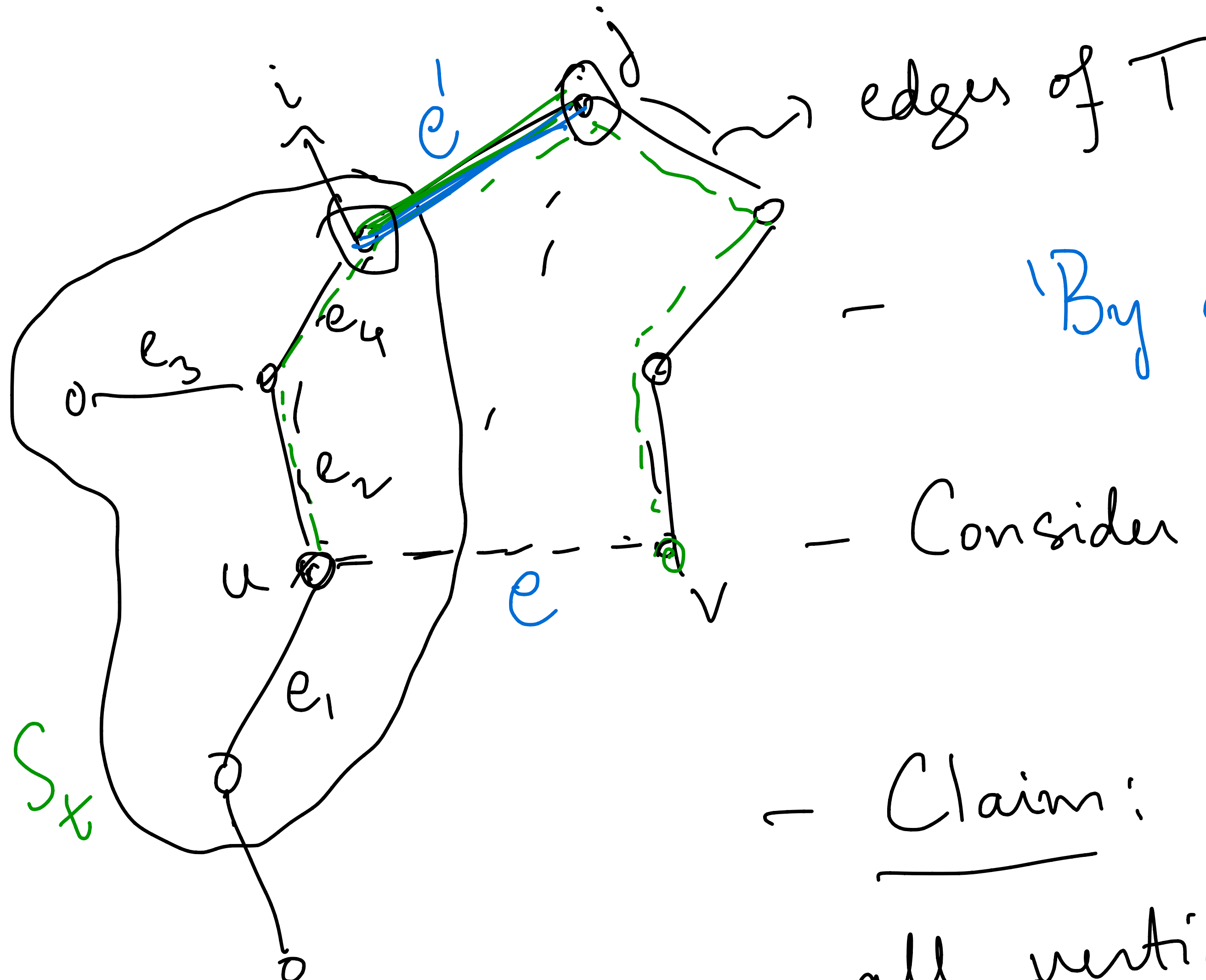


- e has the least wt of all the edges going out of S_t.
- Know: $\exists$ an MST that contains $\{e_1, e_2, e_3, e_4\}$; will show: $\exists$ an MST (for full graph) that contains $\{e_1, \ldots, e_4, e\}$.

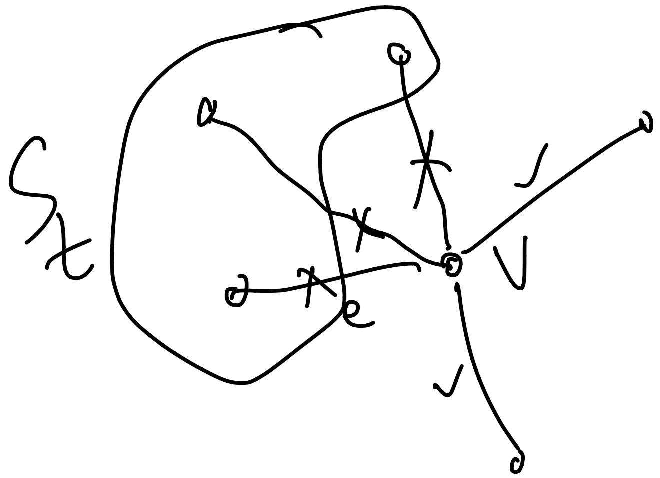
Strategy: Take tree T that contains $\{e_1, \ldots, e_4\}$ and "modify" it to include e.

$u \to v_1 \to v_2 \to \cdots \to v_r \to i \to j \to w_1 \to \cdots \to w_s \to v \qquad u \to v$

Proof of "opt prefix" property



- "By choice, $wt(e) \leq wt(e')$.
- Consider $T' = T \setminus \{e'\} \cup \{e\}$.
- Claim: T' is a spanning tree (i.e., all vertices in G are still connected).
- To show this, it suffices to prove that T' has a path from $i \to j$. (& this is clear — from picture).



$O(|E| + |V| \log n) \longleftarrow O\left(\log n \left(|E| + \underbrace{|V|}_{= n}\right)\right)$

Running time

- Need to maintain S_t & edges going out of S_t.
  - S_t → list.
  - edges going out of S_t → store as priority queue

→ <u>Update</u>:
- pick least wt edge out of S_t $(u \to v)$. — $\log n$
- add end-pt v to S_t — $O(1)$.
- ☆ update edges going out of S_{t+1}.

→ <u>Can see</u>: ☆ takes time $= \deg(v) \cdot \log n$.

Overall run time $= \log n \cdot \left(\sum_v (\deg(v) + 1)\right)$

Minimum spanning tree

- Simple algorithms — analysis slightly tricky
- <u>Common inductive approach for greedy algorithms:</u> show that

  there's an optimal solution that agrees with all choices so far
- Can be solved in O((m + n) log n) time
- Procedure closely related to shortest paths — Dijkstra's algorithm

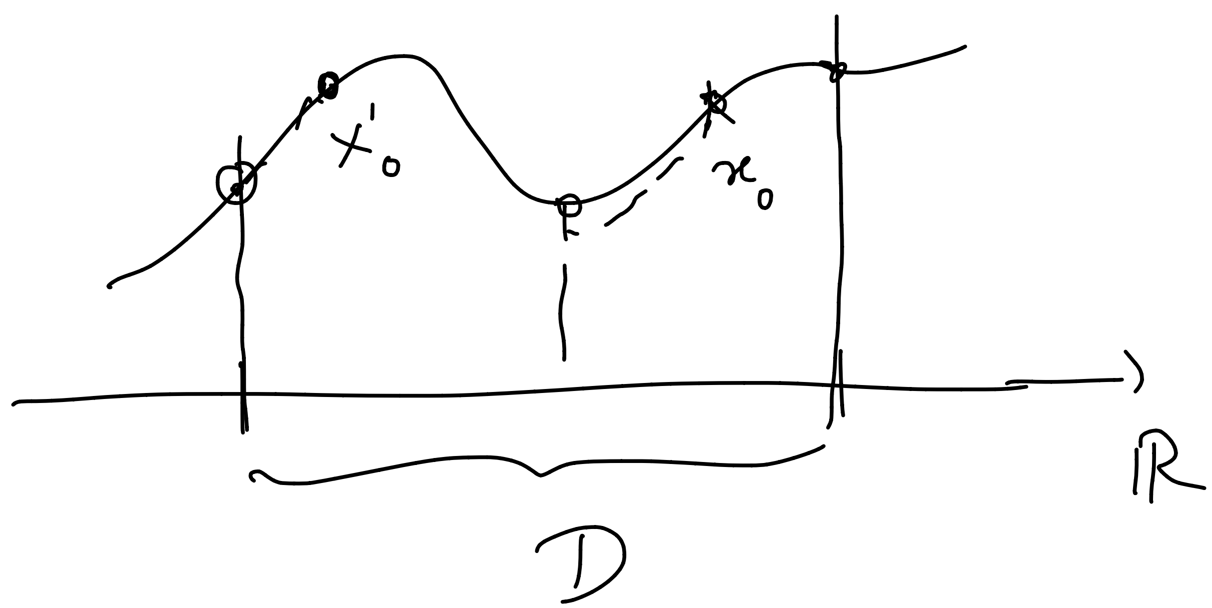
Local search

Main idea

- Start with *any* solution, try improving by moving to “nearby” solution
- Stop if no nearby solution is better

Classic example – function opt

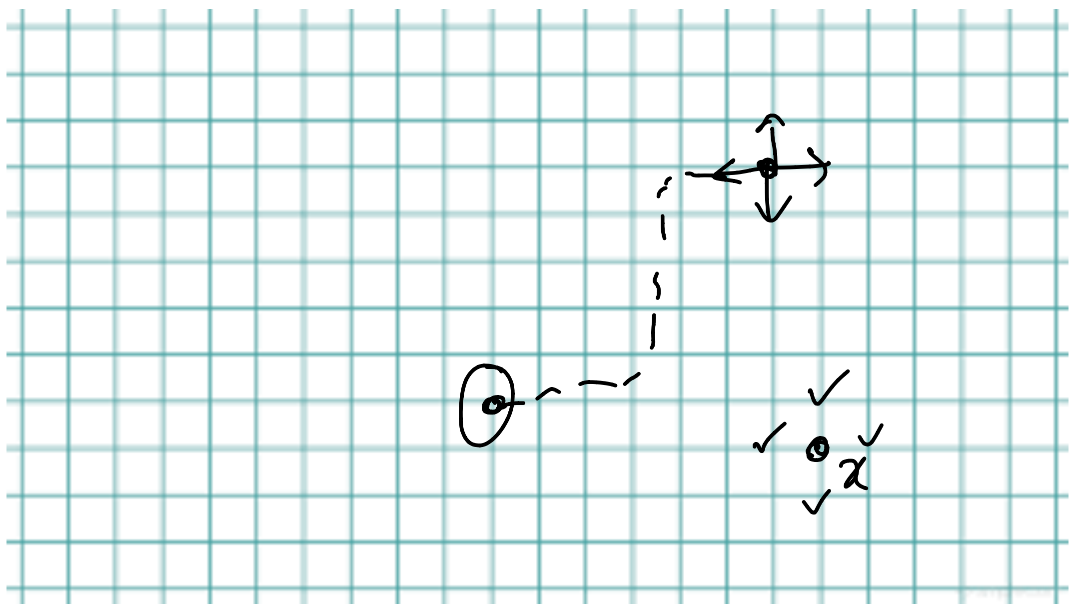
Problem: Let f(x) be a function defined on domain D. Find $\text{argmin}_x f(x)$



Multi-variate functions

$f(x_1, x_2, x_3, \ldots, x_d)$

Grid search



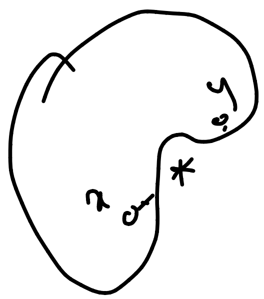
$f : \mathbb{R}^d \longrightarrow \mathbb{R}$.

When is it optimal?

- Any *local optimum* is actually “global” optimum (opt over domain)
- Does this property hold for some natural class?

Statement is not generally true.

Minimizing a convex function



(over all of R^n)

$f : \mathbb{R}^n \to \mathbb{R}$.

Convexity:

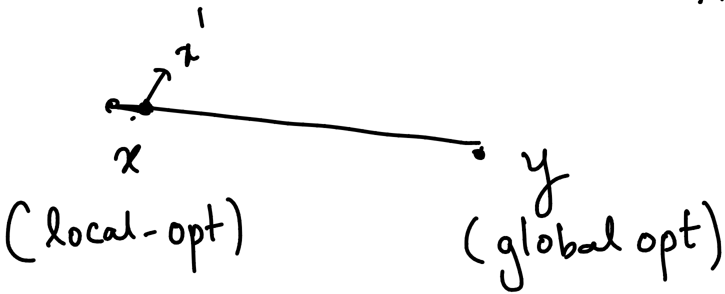
- single-variable fn: $f''(x) > 0 \quad \forall x \in \text{domain}$
- multi-variate : $\nabla^2 f|_x \succcurlyeq 0$. (psd.).
- $\forall x, y \in \text{domain}$. $f\left(\frac{x+y}{2}\right) \leq \frac{f(x)+f(y)}{2}$.

$$f(tx + (1-t)y) \leq t \cdot f(x) + (1-t) f(y)$$

for any $0 \leq t \leq 1$.

Well-known: Let f be a convex fn on $\mathbb{R}^n$. Then any local opt of f is also a global opt!

Suppose $f(y) < f(x)$



$f(x') < f(x)$

Gradient descent

(all of modern ML)

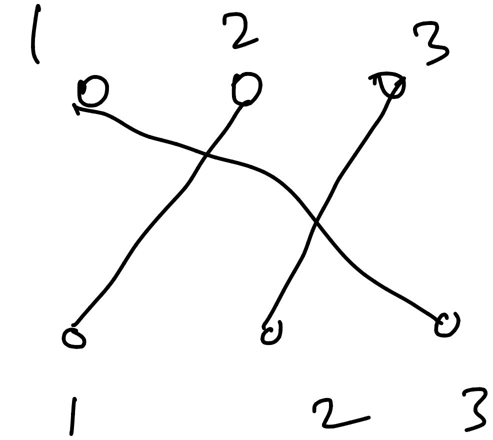
- What is a direction in which function value drops?
- General algorithm:

– start with some x_0

– update $x_{t+1} = x_t - \eta \cdot \nabla f(x_t)$.

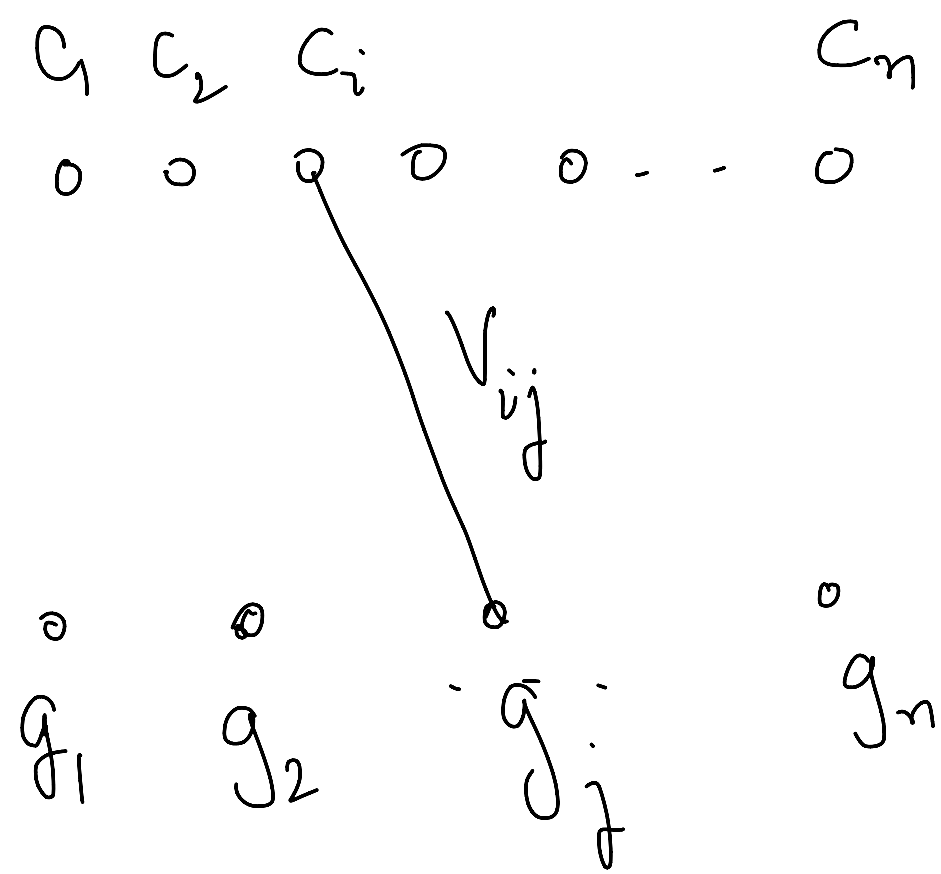
$V_{13} + V_{21} + V_{32}$

Matching problem

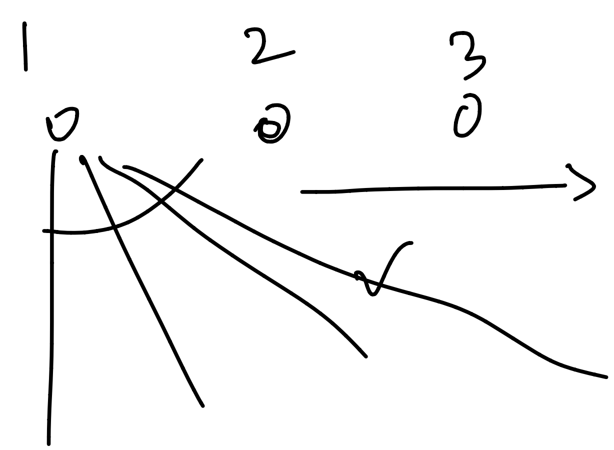


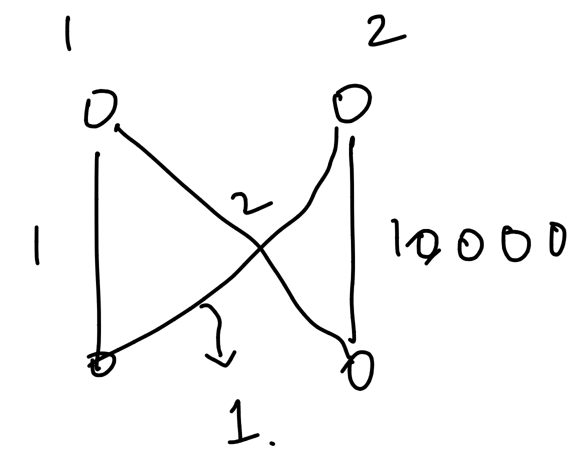
Problem: suppose we have *n* children and *n* gifts. Each child has some "happiness value" (V_{ij}) for each gift. Find an allocation (one gift per child) so that total happiness is maximized.

$V_{ij} \geq 0$ (given).



Matching – greedy?

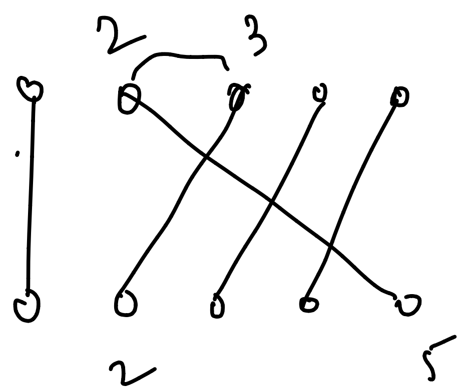




- greedy will not work well.

- Can actually be arbitrarily bad.

Local search (?)



See if swapping gifts of 2, 3 improves solution.

- Candidate local search: for every pair $\{i, j\}$, See if swapping gifts of children i, j improves total cost.

Local search

Claim: take any solution S in which swaps do not increase value. Then total happiness of S >= (1/2) total happiness of OPT solution

2 approximation – proof

Claim: take any solution S in which swaps do not increase value. Then total happiness of S >= (1/2) total happiness of OPT solution