universität freiburg

Algorithm Theory – WS 2024/25

Chapter 2 : Greedy Algorithms 2
(Exchange Arguments, MST, Matroids)

Fabian Kuhn
Dept. of Computer Science
Algorithms and Complexity

Exchange Argument

- General approach that often works to analyze greedy algorithms

- Start with any solution
- Define basic exchange step that allows to transform solution into a new solution that is not worse
- Show that exchange step move solution closer to the solution produced by the greedy algorithm
- Number of exchange steps to reach greedy solution should be finite…

Another Exchange Argument Example

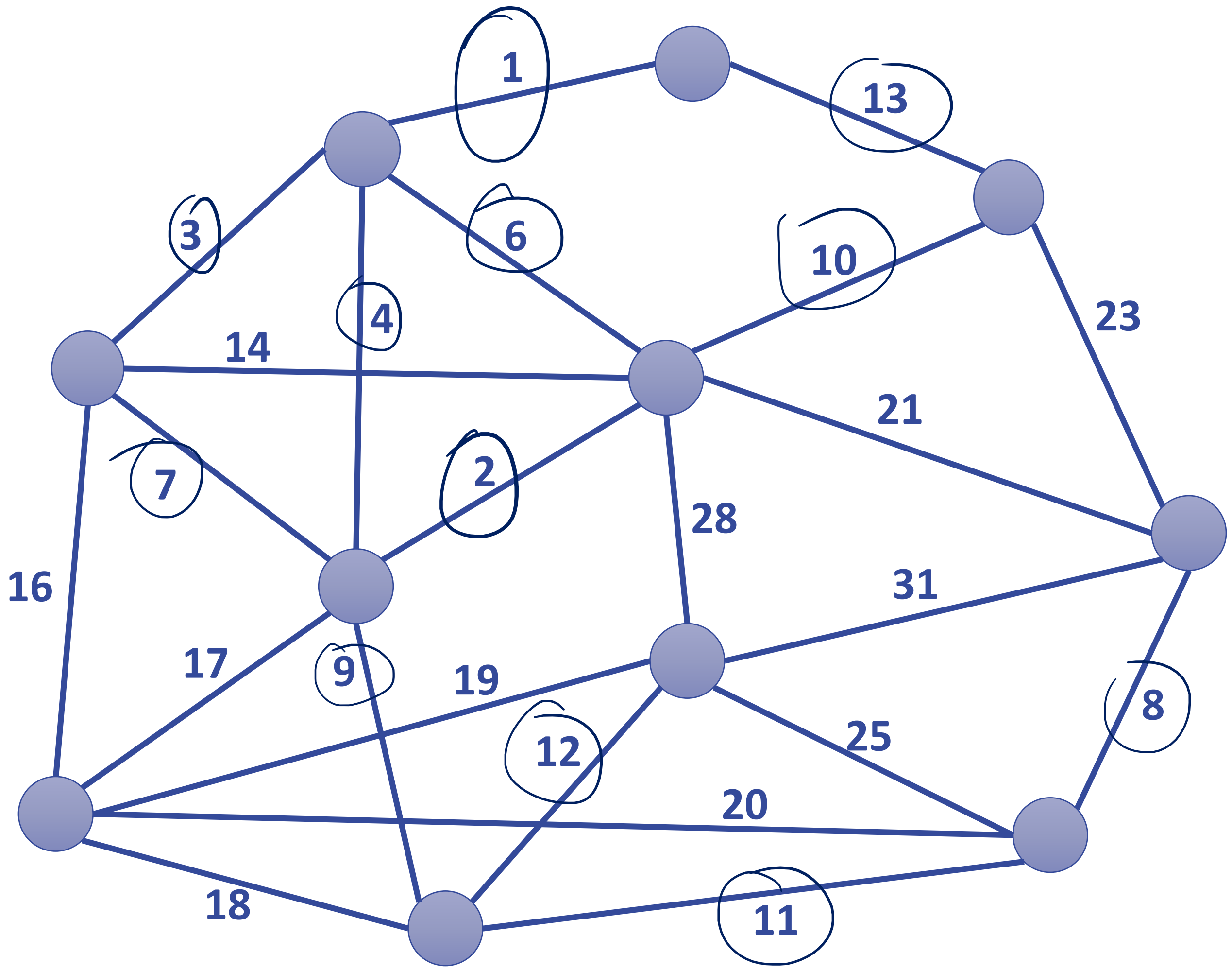
- **Minimum spanning tree (MST)** problem
  - Classic graph-theoretic optimization problem

- **Given**: weighted graph
- **Goal**: spanning tree with min. total weight

- Several greedy algorithms work

- Kruskal's algorithm:
  - Start with empty edge set
  - As long as we do not have a spanning tree:
    add minimum weight edge that doesn't close a cycle

Kruskal Algorithm: Example



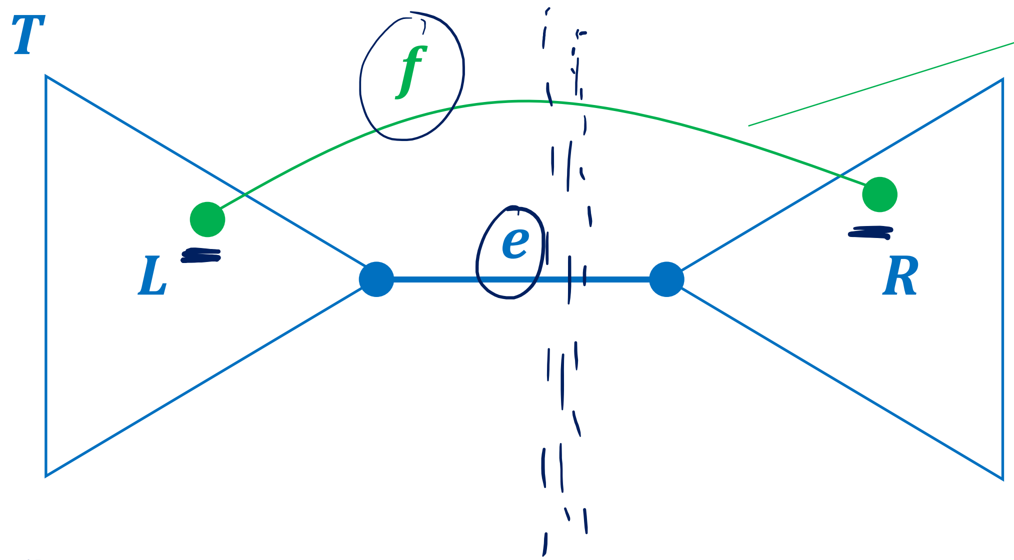
Kruskal is Optimal

Kruskal tree

- Basic exchange step: swap to edges to get from tree T to tree T_K
  - Swap out edge not in Kruskal tree T_K, swap in edge in Kruskal tree T_K
  - Swapping does not increase total weight
- For simplicity, assume, weights are unique

T : any spanning tree T_K: Kruskal tree

Assume that $T \neq T_K \implies \exists e \in T \setminus T_K$



f is the lightest edge connecting L and R

$\implies f \in T_K \setminus T$
$\implies w(f) < w(e)$

$T' \coloneqq T \setminus \{e\} \cup \{f\}$ is a spanning tree of smaller total weight than T.

Matroids

- Same as MST, but more abstract…

Matroid: pair (E, I) — set system

- E: finite set, called the **ground set** — set of elements
- I: finite family of finite subsets of E (i.e., $I \subseteq 2^E$), called **independent sets**

(E, I) needs to satisfy 3 properties:

1. Empty set is independent, i.e., $\emptyset \in I$ (implies that $I \neq \emptyset$)
2. **Hereditary property**: For all $A \subseteq E$ and all $A' \subseteq A$,

   if $A \in I$, then also $A' \in I$

3. **Augmentation / Independent set exchange property:**
   If $A, B \in I$ and $|A| > |B|$, there exists $x \in A \setminus B$ such that

   $$\mathbf{B'} := B \cup \{x\} \in I$$

Simple example:

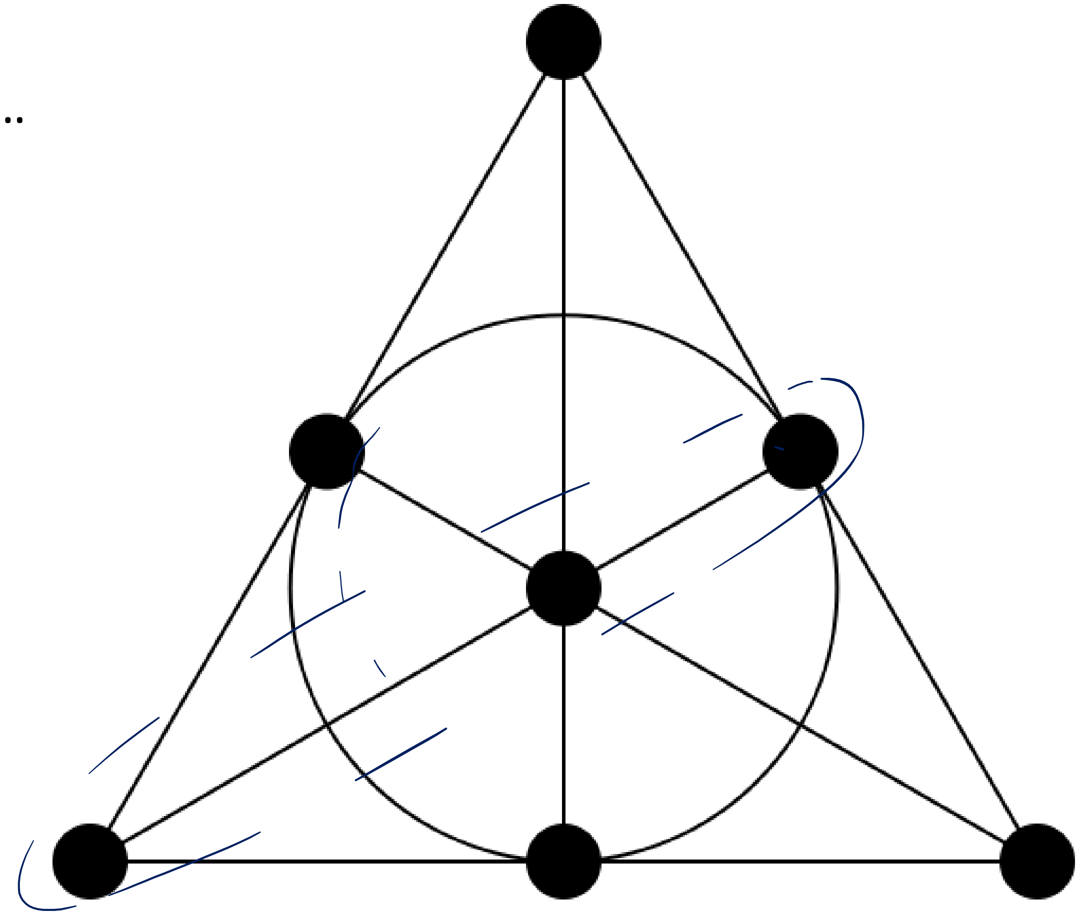
$E := \{1, 2, 3, 4\}$

$$I := \left\{ \begin{matrix} \emptyset, \{1\}, \{2\}, \{3\}, \{4\}, \{1,2\}, \{1,3\}, \\ \{1,4\}, \{2,3\}, \{2,4\}, \{3,4\} \end{matrix} \right\}$$

$A = \{2, 4\}$, $B = 3$

Example

- Fano matroid:
  - Smallest finite projective plane of order 2…



Matroids and Greedy Algorithms

Weighted matroid: each $e \in E$ has a weight $w(e) \geq 0$

- Recall that all independent sets in I consist of a finite set of elements of E.

Goal: find maximum weight independent set

Greedy algorithm:

1. Start with $S = \emptyset$
2. Add max. weight $x \in E \setminus S$ to S such that $S \cup \{x\} \in I$

Claim: greedy algorithm computes optimal solution

Greedy is Optimal

Matroid (E, I),
weights $w(x) \geq 0$ for all $x \in E$

S: greedy solution A: any other solution (ind. set)

$S \subseteq E$, $S \in I$ $A \subseteq E$, $A \in I$

$|S| \geq |A|$: [$s = |S|$, $a = |A|$, $s \geq a$]

for contradiction, assume that $|A| > |S|$: exch. prop: $\exists x \in A \setminus S$ s.t. $S \cup \{x\} \in I$

greedy would have added x

$S' \subseteq S$
$S' \cup \{x\} \subseteq S \cup \{x\}$

$w(S) \geq w(A)$:

for contradiction, assume $w(S) < w(A)$

$S = \{x_1, x_2, \dots, x_s\}$ $w(x_1) \geq w(x_2) \geq \dots \geq w(x_s)$

$A = \{y_1, y_2, \dots, y_a\}$ $w(y_1) \geq w(y_2) \geq \dots \geq w(y_a)$

We will show:

$\forall i \in \{1, \dots, a\}$: $w(x_i) \geq w(y_i)$ $(*)$

$\hookrightarrow w(S) \geq w(A)$
$\hookrightarrow s \geq a$

$\neg(*) \Longrightarrow$ there is a smallest index $k \leq a$: $w(x_k) < w(y_k)$

$S' = \{x_1, x_2, \dots, x_{k-1}\}$ exch. prop: $\exists y \in A' \setminus S'$ s.t. $S' \cup \{y\} \in I$

$A' = \{y_1, y_2, \dots, y_k\}$ $w(y) \geq w(y_k) > w(x_k)$

greedy considers y before x_k

$|A'| > |S'|$ greedy would add y $\lightning$

Matroids: Examples

Forests of a graph $G = (V, E)$:

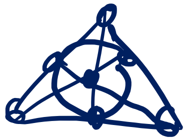
- forest F: subgraph with no cycles (i.e., $F \subseteq E$)
- $\mathcal{F}$: set of all forests → $(E, \mathcal{F})$ is a matroid
- Greedy algorithm gives maximum weight forest
  - equivalent to MST problem

Bicircular matroid of a graph $G = (V, E)$:

- $\mathcal{B}$: set of edges such that every connected subset has ≤ 1 cycle
- $(E, \mathcal{B})$ is a matroid → greedy gives max. weight such subgraph

Linearly independent vectors:

- Vector space V, E: finite set of vectors, I: sets of lin. indep. vect.
- Fano matroid can be defined like that



Forest Matroid of Graph $G = (V, E)$

Ground set: E (edges) **Independent sets:** $\mathcal{F}$ (forests of G)

Basic properties: $\emptyset \in \mathcal{F}$ + hereditary property

- Empty graph has no cycles, removing edges doesn't create cycles

Independent set exchange property:

- Given $\mathcal{F}_1, \mathcal{F}_2$ s.t. $|\mathcal{F}_1| > |\mathcal{F}_2|$
  - $\exists e \in \mathcal{F}_1 \setminus \mathcal{F}_2$ s.t. $\mathcal{F}_2 \cup \{e\}$ is a forest
- $\mathcal{F}_1$ needs to have an edge e connecting two components of $\mathcal{F}_2$
  - Because it can only have $|\mathcal{F}_2|$ edges connecting nodes inside components