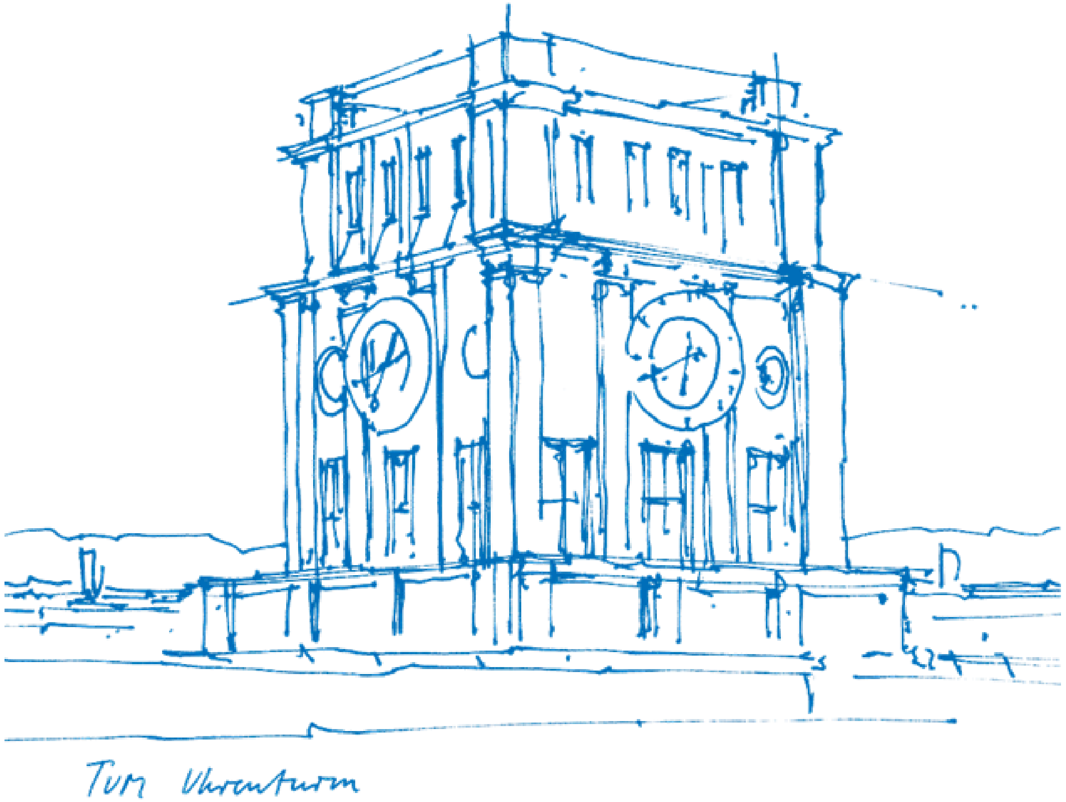
AACPP 2025

Week 12: The End

Mateusz Gienieczko, Mykola Morozov

School of Computation, Information and Technology
Technical University of Munich

2025.07.22



Course survey

APD – All Paws on Deck

Given a graph of pairwise animosities determine if we can take k cats such that none of them dislike each other.

This is asking for Maximal Independent Set of size at least k.

APD – All Paws on Deck

Key observation – k is very very large, so we should look for the dual structure – the Minimal Cover.

Set $l = n - k$. Then we have $l \leq 15$, which gives some hope of a good parameterised complexity.

APD – All Paws on Deck

Let’s make a backtrack first, since we’ll need one in the model anyway.

Min Cover is solved simply – either take a vertex and remove all its edges or don’t, stop if you run out of vertices.

This works in $\mathcal{O}\left(\binom{n}{l}(n+m)\right) = \mathcal{O}\left(\frac{n^l m}{l!}\right)$, but it should work for Subtask 1.

APD – All Paws on Deck

A much better backtrack can be achieved with a simple observation: **if we do not take v into the cover, we need to take all $N(v)$ into the cover**.

So at each step:

- if a vertex v is isolated: ignore it;
- otherwise: remove all edges from v;
- branch: take the vertex to the cover;
- branch: don't take the vertex, take all its neighbourhood.

This limits the depth of the backtrack tree by l, giving $\mathcal{O}\left((n+m)2^l\right)$.

APD – All Paws on Deck

Min Cover is a classic kernelisation problem.

Observation: if a vertex has degree $> l$ then it has to be in any valid vertex cover of size at most l.

For kernelisation apply the two rules:

1. Remove all isolated vertices from the graph,
2. If a vertex with degree $> l$ exists, take it to the cover, remove it from the graph, and decrease l by one.

If a cover of l vertices exists in this graph then the graph can have at most l^2 edges. Since there are no isolated vertices, then also $l(l + 1)$ vertices.

APD – All Paws on Deck

In other words:

1. Apply kernelisation.
2. If we run out of vertices (exceed l) then the answer is `CATASTROPHE`.
3. Otherwise, we end up with a graph with $|V|, |E| = \mathcal{O}(l^2)$. Run the backtrack.

Kernelisation rules can be applied quickly: keep track of degrees of all vertices, go through all those that initially had degree $> l$. Isolated vertices can be removed at the end. Total $\mathcal{O}(n + m)$.

Backtrack on the reduced graph takes $\mathcal{O}\left(l^2 2^l\right)$ for a total of $\mathcal{O}\left(n + m + l^2 2^l\right)$.

What We Covered

- Greedy and dynamic programming (DP)
- Trees
- Graphs
- Ways to turn graphs into trees (DFS, BFS, Dijkstra, MST)
- Ways to run DP on graphs (Toposort)
- Advanced graph algorithms (Matchings, flows)
- Binary Search Trees
- Number theory
- String algorithms (KMP, tries, suffix tables)
- Some problems can't* even be solved efficiently (NP-completeness)

What We Didn't Cover

- Suffix trees (see slides of Lecture 9)
-
-
-
-
-
-
-
-
-
-

What We Didn’t Cover

- Suffix trees (see slides of Lecture 9)
- Matrix operations, Strassen, numerical methods
-
-
-
-
-
-
-
-
-

What We Didn't Cover

- Suffix trees (see slides of Lecture 9)
- Matrix operations, Strassen, numerical methods
- Combinatorics (e.g. binomial, Catalan numbers)
-
-
-
-
-
-
-
-

What We Didn’t Cover

- Suffix trees (see slides of Lecture 9)
- Matrix operations, Strassen, numerical methods
- Combinatorics (e.g. binomial, Catalan numbers)
- Computational geometry
-
-
-
-
-
-
-

What We Didn't Cover

- Suffix trees (see slides of Lecture 9)
- Matrix operations, Strassen, numerical methods
- Combinatorics (e.g. binomial, Catalan numbers)
- Computational geometry
- 2-SAT, tree colouring, heavy-light decomposition
-
-
-
-
-
-

What We Didn’t Cover

- Suffix trees (see slides of Lecture 9)
- Matrix operations, Strassen, numerical methods
- Combinatorics (e.g. binomial, Catalan numbers)
- Computational geometry
- 2-SAT, tree colouring, heavy-light decomposition
- Max-flow min-cost, cycle cancelling, push-relabel
-
-
-
-
-

What We Didn’t Cover

- Suffix trees (see slides of Lecture 9)
- Matrix operations, Strassen, numerical methods
- Combinatorics (e.g. binomial, Catalan numbers)
- Computational geometry
- 2-SAT, tree colouring, heavy-light decomposition
- Max-flow min-cost, cycle cancelling, push-relabel
- Matching in general graphs (Edmond’s algorithm)
-
-
-
-

What We Didn’t Cover

- Suffix trees (see slides of Lecture 9)
- Matrix operations, Strassen, numerical methods
- Combinatorics (e.g. binomial, Catalan numbers)
- Computational geometry
- 2-SAT, tree colouring, heavy-light decomposition
- Max-flow min-cost, cycle cancelling, push-relabel
- Matching in general graphs (Edmond’s algorithm)
- Linear programming
-
-
-

What We Didn’t Cover

- Suffix trees (see slides of Lecture 9)
- Matrix operations, Strassen, numerical methods
- Combinatorics (e.g. binomial, Catalan numbers)
- Computational geometry
- 2-SAT, tree colouring, heavy-light decomposition
- Max-flow min-cost, cycle cancelling, push-relabel
- Matching in general graphs (Edmond’s algorithm)
- Linear programming
- Game theory
-
-

What We Didn’t Cover

- Suffix trees (see slides of Lecture 9)
- Matrix operations, Strassen, numerical methods
- Combinatorics (e.g. binomial, Catalan numbers)
- Computational geometry
- 2-SAT, tree colouring, heavy-light decomposition
- Max-flow min-cost, cycle cancelling, push-relabel
- Matching in general graphs (Edmond’s algorithm)
- Linear programming
- Game theory
- Approximations for NP-complete problems
-

What We Didn't Cover

- Suffix trees (see slides of Lecture 9)
- Matrix operations, Strassen, numerical methods
- Combinatorics (e.g. binomial, Catalan numbers)
- Computational geometry
- 2-SAT, tree colouring, heavy-light decomposition
- Max-flow min-cost, cycle cancelling, push-relabel
- Matching in general graphs (Edmond's algorithm)
- Linear programming
- Game theory
- Approximations for NP-complete problems
- ... probably even more!

Grades

grade	points
1.0	292
1.3	246
1.7	227
2.0	204
2.3	191
2.7	181
3.0	171
3.3	161
3.7	156
4.0	151
4.3	101
4.7	51

If you haven't signed up on the official spreadsheet before today, then you will not receive a grade…

… unless you send a *really* polite email to us that explains how that happened.

See you around!