§4 Computational Complexity

KAIST
CS422 M. Ziegler

- Model of computation *with* (bit) cost:
  WHILE+
- Complexity of Arithmetic
- Complexity classes
  $\mathcal{P}, \mathcal{NP}, \mathcal{PSPACE}, \mathcal{EXP}$
- and their inclusion relations
- Encoding graphs/non-integer data
- Example problems:
  3COL, EC, HC, VC, ILP, IS, Clique

Model of Computational Cost

KAIST
CS422 M. Ziegler

WHILE takes expon. time to add two n-bit integers

Now **WHILE+** programs: Input $x_1 \in \mathbb{N}$, output $x_0 \in \mathbb{N}$

$$x_j := 0 \mid x_j := 1 \mid x_j := x_i + x_k \mid x_j := x_i - x_k \mid$$
$$x_j := x_i \div 2 \mid \quad P;P \mid \text{WHILE } x_i \text{ DO } P \text{ END}$$

Definitions: binary *length* of $x \in \mathbb{N}$: $\ell(x) = \lceil \log_2(1+x) \rceil$

- time of a WHILE+ program P on input $\underline{x}=(x_1,\ldots x_k)$
- space (=memory) used: $\max_t \ell(\underline{x}) := \ell(x_1)+\ldots+\ell(x_k)$
- asymptotic time/space $t(n)/s(n)$:
  worst-case over all inputs $\underline{x}$ with $\ell(\underline{x})<n$
- Recall pairing function $\langle x,y \rangle := x + (x+y)\cdot(x+y+1)/2$

Complexity of Arithmetic

WHILE takes expon. time to add two n-bit integers

Now WHILE+ programs: Input $x_1 \in \mathbb{N}$, output $x_0 \in \mathbb{N}$

$x_j := 0 \mid x_j := 1 \mid x_j := x_i + x_k \mid x_j := x_i - x_k \mid$
$x_j := x_i \div 2 \mid$ $P;P \mid \text{WHILE } x_i \text{ DO } P \text{ END}$

Multiplication by repeated addition: expon. time

Long multiplication: linear time

Long division: linear time

Un-/pairing: linear time

$\ell(\langle x,y \rangle) = \Theta(\ \ell(x) + \ell(y)\ )$

$\ell(x) = \Theta(\log x)$

- asymptotic time/space $t(n)$/$s(n)$: worst-case over all inputs $\underline{x}$ with $\ell(\underline{x})<n$
- Recall pairing function $\langle x,y \rangle := x + (x+y)\cdot(x+y+1)/2$

Preliminaries: Graphs and Coding

- A *directed* graph $G=(V,E)$ is a finite set V of *vertices* and a set $E \subseteq V \times V$ of *edges*
- Call G *undirected* if it holds $(u,v) \in E \Leftrightarrow (v,u) \in E$
- sometimes $c:E \to \mathbb{N}$ assigning *weights* to edges.

For input to a WHILE+ program:

- Represent (G,c) as adjacency matrix $A \in \mathbb{N}^{V \times V}$
  - $A[u,v] := c(i,j)$ for $(u,v) \in E$,
  - $A[u,v] :=$ "∞" for $(u,v) \notin E$
- Undirected case: only upper triangular matrix.
- Encoding $\langle G,c \rangle \in \mathbb{N}$ has $|V| \leq |\langle G,c \rangle| \leq O(|V|^2 \cdot \log |c|_\infty)$

Some Complexity Classes

Definition: a) A WHILE+ program computes the function $f:\mathbb{N}\to\mathbb{N}$ if on input x it prints $f(x)$ and terminates in time $t(n)$ / space $s(n)$, $n:=\ell(\underline{x})$

Polynom.growth: $\exists k\ t(n)\leq O(n^k)$; exponential: $2^{O(n^k)}$

Def: For decision problems $L\subseteq\mathbb{N}$ or $L\subseteq\{0,1\}^*$

- $\mathcal{P}$ = { L decidable in polynomial time }
- $\mathcal{NP}$ = { L verifiable in polynomial time }, i.e.

$L=\{\ x\in\mathbb{N} : \exists y\in\mathbb{N},\ \ell(y)\leq\text{poly}(\ell(x)),\ \langle x,y\rangle\in V\ \},\ V\in\mathcal{P}$

- $\mathcal{PSPACE}$ = { L decidable in polynomial space }
- $\mathcal{EXP}$ = { L decidable in exponential time }

Theorem: $\mathcal{P}\subseteq\mathcal{NP}\subseteq\mathcal{PSPACE}\subseteq\mathcal{EXP}$

Example Problem (0)

Def: A 3-coloring of $G=(V,E)$ is a mapping $\gamma:V\to\{R,G,B\}$ s.t. $\gamma(u)\neq\gamma(v)$ for every $(u,v)\in E$.

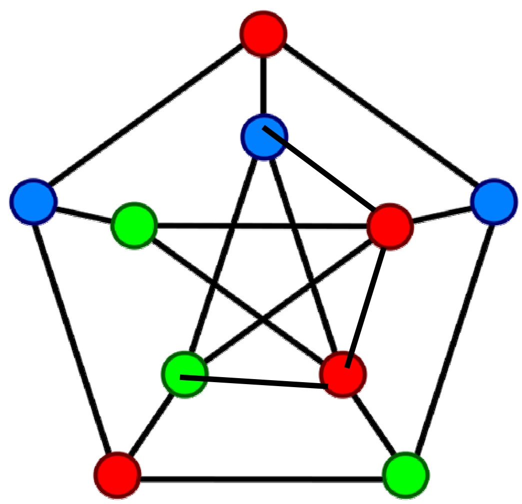
Examples:

a) The *Petersen Graph* admits a 3-coloring.

b) This graph, too.

c) This one still.

d) But not this one.



$x=\langle G\rangle,\ y=\langle\gamma(1),\ldots\gamma(|V|)\rangle$

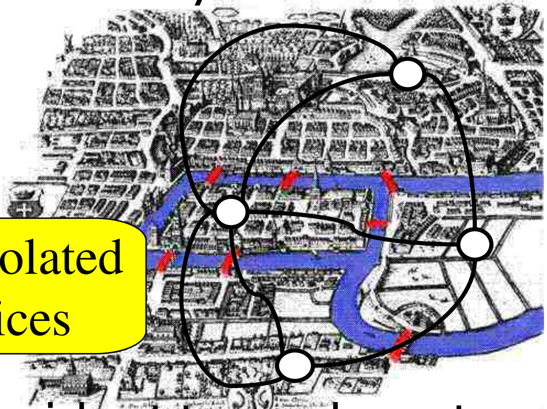
3COL= { $\langle G\rangle$ | G admits a 3-coloring } $\in\mathcal{NP}$

$\mathcal{NP}\ni\{x\in\mathbb{N}: \exists y,\ \ell(y)\leq\text{poly}(\ell(x)),\ \langle x,y\rangle\in V\},\ V\in\mathcal{P}$

Example Problems (I)

In an undirected graph G, Eulerian cycle traverses each edge precisely once;

Hamiltonian cycle visits each vertex precisely once.



G admitting a Eulerian cycle is connected and has an even number of edges incident to each vertex

save isolated vertices

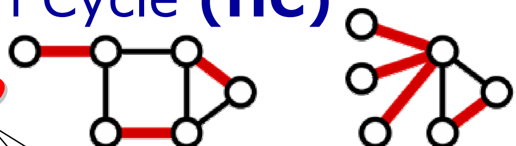
Theorem: Conversely every connected graph with an even number of edges incident to each vertex admits a Eulerian cycle.

EC := { $\langle G\rangle$ | G has a Eulerian cycle} $\mathcal{P}$

HC := { $\langle G\rangle$ | G has Hamiltonian cycle} $\mathcal{NP}$

Example Problems (II)

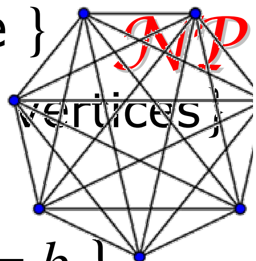
- Eulerian **(EC)** vs. Hamiltonian Cycle **(HC)**
- (Minimum) **Edge Cover** $\mathcal{P}$



"To graph G, find a smallest subset F of edges s.t. any vertex v is adjacent to at least one $e\in F$."

Greedily extend a maximum matching

- vs. **Vertex Cover (VC)** $\mathcal{NP}$
- **CLIQUE** = { $\langle G,k\rangle$ | G contains a k-clique } $\mathcal{NP}$
- **IS**={$\langle G,k\rangle$: G has k pairwise non-adjacent vertices}



- Integer Linear Programming $\mathcal{NP}$?

ILP = { $\langle \underline{A},\underline{b}\rangle$: $\underline{A}\in\mathbb{Z}^{n\times m}$, $\underline{b}\in\mathbb{Z}^m$, $\exists\underline{x}\in\mathbb{Z}^n$: $\underline{A}\cdot\underline{x}=\underline{b}$ }

VC = {$\langle V,E,k\rangle$: $\exists U\subseteq V$, $|U|=k$, $\forall(x,y)\in E$: $x\in U \vee y\in U$ }

$\mathcal{NP} \ni$ {$x\in\mathbb{N}$: $\exists y$, $\ell(y)\leq\text{poly}(\ell(x))$, $\langle x,y\rangle\in V$ }, $V\in\mathcal{P}$

Example Problems (III)

Def: A Boolean term $\Phi(Y_1,\ldots Y_n)$ is composed from variables $Y_1,\ldots Y_n$, constants 0 and 1, and operations $\vee$, $\wedge$, $\neg$.

Examples:
- 0
- $(\neg x \vee y) \wedge (x \vee \neg y)$
- $(\neg x \vee y) \wedge (x \vee y) \wedge \neg y$
- $(\neg x \vee y) \wedge (x \vee \neg z) \wedge (z \vee \neg y) \wedge x \wedge (\neg y)$

literals

clause

Φ in 3-CNF if $\Phi = \bigwedge\big((\neg)y_i \vee (\neg)y_j \vee (\neg)y_\ell\big)$

EVAL: Given $\langle\Phi(Y_1,\ldots Y_n)\rangle$ and $y_1,\ldots y_n \in \{0,1\}$, does $\Phi(y_1,\ldots y_n)$ evaluate to 1 ? $\in \mathcal{P}$

[*k*-] SAT: Given $\Phi(Y_1,\ldots Y_n)$ [in *k*-CNF], does it hold $\exists y_1,\ldots y_n \in \{0,1\}: \Phi(y_1,\ldots y_n)=1$?

Non-Deterministic WHILE+

Theorem: $L \subseteq \mathbb{N}$ is accepted by a *non*-deterministic polynomial-time WHILE+ program iff $L \in \mathcal{NP}$.

$$x_j := 0 \mid x_j := 1 \mid x_j := x_i + x_k \mid x_j := x_i - x_k \mid$$
$$x_j := x_i \div 2 \mid \text{guess } x_j \mid P;P \mid \text{WHILE } x_i \text{ DO } P \text{ END}$$

Definition: A *non*-deterministic WHILE+ program may (repeatedly) guess a bit (0/1).

- Its runtime is $\leq t(n)$ if it makes no more than $t(\ell(x_1))$ steps, regardless of the guesses.
- It accepts input x_1 if there exists some choice of guessed values such as to return $x_0=1$.
- It rejects x_1 if no choice of guesses returns $x_0=1$.