

Foundations of Complexity Theory

Lecture 2: Foundational Relations I

October 21, 2026

Next Dates

LECTURE (WED 1:15PM)	
Oct 21st	Lecture 2: Fundamental relations I
28th	Lecture 3: Fundamental relations II
Nov 4th	Lecture 4: Fundamental relations III
11th	Lecture 5: Reductions & complete problems I
18th	
(public holiday)	
25th	Lecture 6: Complete problems II
Dec 2nd	
(dies academicus)	
9th	Lecture 7: NP-complete problems

LECTURE (WED 1:15PM)	
Dec 16th	Lecture 8: Complete problems III
23rd	
30th	
Jan 6th	Lecture 9: Boolean circuits I
13th	Lecture 10: Boolean circuits II
20th	Lecture 11: Razborov's theorem I
27th	
Feb 3rd	Lecture 12: Razborov's theorem II

Contents

- 1 Basic complexity classes & their relations
- 2 Reductions & complete problems
- 3 Boolean circuits & Razborov's theorem
- 4 Randomization & interactive proof systems
- 5 Polynomial primality test

Complexity Classes

Standard classes

$$\mathbf{L} = \mathbf{DSpace}(\log x)$$

$$\mathbf{NL} = \mathbf{NSpace}(\log x)$$

$$\mathbf{P} = \bigcup_{i \in \mathbb{N}_+} \mathbf{DTIME}(x^i)$$

$$\mathbf{NP} = \bigcup_{i \in \mathbb{N}_+} \mathbf{NTIME}(x^i)$$

$$\mathbf{PSPACE} = \bigcup_{i \in \mathbb{N}_+} \mathbf{DSpace}(x^i) \stackrel{\dagger}{=} \bigcup_{i \in \mathbb{N}_+} \mathbf{NSpace}(x^i)$$

$$\mathbf{EXPTIME} = \bigcup_{i \in \mathbb{N}_+} \mathbf{DTIME}(2^{x^i})$$

$$\mathbf{NEXPTIME} = \bigcup_{i \in \mathbb{N}_+} \mathbf{NTIME}(2^{x^i})$$

$\dagger$ = Savitch's theorem

§2.1 Definition (Computed function)

Let $M = (Q, \Sigma, \Gamma, \Sigma', \delta, \square, q_0, q_+, q_-)$ deterministic TM.

M **computes** partial function $T_M: \Sigma^* \dashrightarrow (\Sigma')^*$

- $T_M(w)$ is output of protocol $(\rho_0, \dots, \rho_n)$ of unique (accepting) computation of M on input w for all $w \in L(M)$
- $T_M(w)$ undefined otherwise

(Corresponds to transformation semantics)

Theorem of Borodin-Trakhtenbrot

Allan Borodin (* 1941)

- US computer scientist
- Professor at University of Toronto
- Proved result in 1972
- Received CRM-Fields-PIMS prize in 2008



© Fields Institute

Boris Trakhtenbrot (* 1921; † 2016)

- Russ. mathematician
- Proved result already 1964
- Emigrated to Israel in 1981



© Cornell University

Theorem of Borodin-Trakhtenbrot

§2.2 Theorem (Theorem of Borodin-Trakhtenbrot)

Let $f: \mathbb{N} \rightarrow \mathbb{N}$ total, computable, and monotonic increasing function with $f(n) \geq n$ for all $n \in \mathbb{N}$. There effectively exists total, computable, and monotonic increasing function $g: \mathbb{N} \rightarrow \mathbb{N}$ with $g(n) \geq n$ for all $n \in \mathbb{N}$ and

$$\text{DTIME}(g) = \text{DTIME}(g ; f)$$

Theorem of Borodin-Trakhtenbrot

§2.2 Theorem (Theorem of Borodin-Trakhtenbrot)

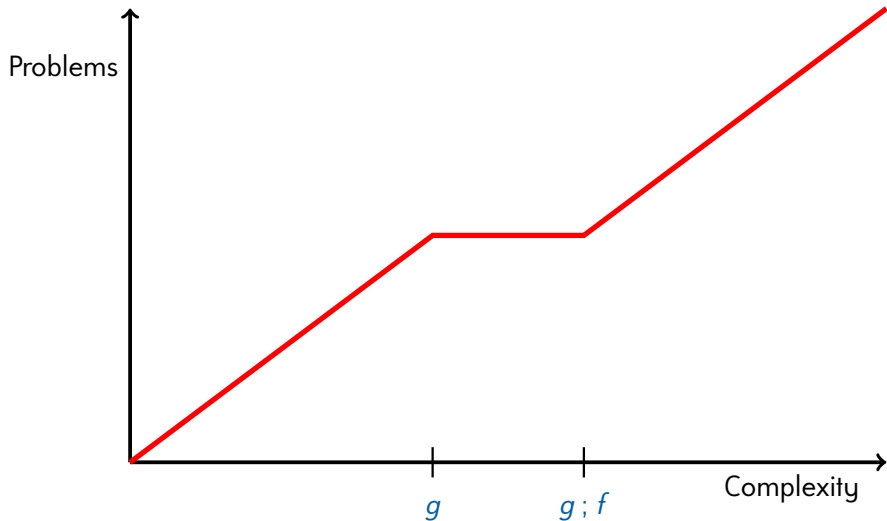
Let $f: \mathbb{N} \rightarrow \mathbb{N}$ total, computable, and monotonic increasing function with $f(n) \geq n$ for all $n \in \mathbb{N}$. There effectively exists total, computable, and monotonic increasing function $g: \mathbb{N} \rightarrow \mathbb{N}$ with $g(n) \geq n$ for all $n \in \mathbb{N}$ and

$$\text{DTIME}(g) = \text{DTIME}(g ; f)$$

Notes

- Known as *gap theorem*
- $(g ; f)(n) = f(g(n))$ for all $n \in \mathbb{N}$
- Effective existence means
given TM for f we can write down TM for g

Theorem of Borodin-Trakhtenbrot



No problems have “minimal” complexity in range $[g, g; f]$

→ gap

Theorem of Borodin-Trakhtenbrot

Proof (1/3)

Let a recursive enumeration of all deterministic TM and
let $t_\ell: \mathbb{N} \rightarrow \mathbb{N} \cup \{\infty\}$ for all $\ell \in \mathbb{N}$ such that for all $m \geq 2$

$$t_\ell(m) = \sup\{n \mid w \in \Sigma^{\leq m-2}, (\kappa_0, \dots, \kappa_n) \text{ computation of } a(\ell) \text{ for } w\}$$

(time consumption of all computations of $a(\ell)$ on inputs up to length $m - 2$)

Theorem of Borodin-Trakhtenbrot

Proof (1/3)

Let a recursive enumeration of all deterministic TM and
let $t_\ell: \mathbb{N} \rightarrow \mathbb{N} \cup \{\infty\}$ for all $\ell \in \mathbb{N}$ such that for all $m \geq 2$

$$t_\ell(m) = \sup\{n \mid w \in \Sigma^{\leq m-2}, (\kappa_0, \dots, \kappa_n) \text{ computation of } a(\ell) \text{ for } w\}$$

(time consumption of all computations of $a(\ell)$ on inputs up to length $m-2$)
 $M_m = \{t_\ell(m) \mid \ell \leq m\}$ is obviously finite subset of $\mathbb{N} \cup \{\infty\}$.

Theorem of Borodin-Trakhtenbrot

Proof (1/3)

Let a recursive enumeration of all deterministic TM and
let $t_\ell: \mathbb{N} \rightarrow \mathbb{N} \cup \{\infty\}$ for all $\ell \in \mathbb{N}$ such that for all $m \geq 2$

$$t_\ell(m) = \sup\{n \mid w \in \Sigma^{\leq m-2}, (\kappa_0, \dots, \kappa_n) \text{ computation of } a(\ell) \text{ for } w\}$$

(time consumption of all computations of $a(\ell)$ on inputs up to length $m-2$)

$M_m = \{t_\ell(m) \mid \ell \leq m\}$ is obviously finite subset of $\mathbb{N} \cup \{\infty\}$.

Let $g(0) = 0$ and $g(1) = 1$. Hence there exists $g(m) \in \mathbb{N}$ for every $m \geq 2$
such that $M_m \cap [g(m), f(g(m))] = \emptyset$.

Theorem of Borodin-Trakhtenbrot

$$t_\ell(m) = \sup\{n \mid w \in \Sigma^{\leq m-2}, (\kappa_0, \dots, \kappa_n) \text{ computation of } a(\ell) \text{ for } w\}$$
$$M_m = \{t_\ell(m) \mid \ell \leq m\}$$

Proof (2/3)

Exists $g(m) \in \mathbb{N}$ for every $m \geq 2$ such that $M_m \cap [g(m), f(g(m))] = \emptyset$.

We compute such $g(m)$ by:

- ① $s \leftarrow \max(m, g(m-1))$
- ② Increase s until $t_\ell(m) \notin [s, f(s)]$ for all $\ell \leq m$
- ③ Return s

Theorem of Borodin-Trakhtenbrot

$$t_\ell(m) = \sup\{n \mid w \in \Sigma^{\leq m-2}, (\kappa_0, \dots, \kappa_n) \text{ computation of } a(\ell) \text{ for } w\}$$
$$M_m = \{t_\ell(m) \mid \ell \leq m\}$$

Proof (2/3)

Exists $g(m) \in \mathbb{N}$ for every $m \geq 2$ such that $M_m \cap [g(m), f(g(m))] = \emptyset$.

We compute such $g(m)$ by:

- 1 $s \leftarrow \max(m, g(m-1))$
- 2 Increase s until $t_\ell(m) \notin [s, f(s)]$ for all $\ell \leq m$
- 3 Return s

Obviously g is total, computable, monotonic increasing, and obeys $g(m) \geq m$ for all $m \in \mathbb{N}$. It remains to prove $\text{DTIME}(g) = \text{DTIME}(g; f)$.

Theorem of Borodin-Trakhtenbrot

$$t_\ell(m) = \sup\{n \mid w \in \Sigma^{\leq m-2}, (\kappa_0, \dots, \kappa_n) \text{ computation of } a(\ell) \text{ for } w\}$$
$$M_m = \{t_\ell(m) \mid \ell \leq m\}$$

Proof (3/3)

Exists $g(m) \in \mathbb{N}$ for every $m \geq 2$ such that $M_m \cap [g(m), f(g(m))] = \emptyset$.

It remains to prove $\text{DTIME}(g) = \text{DTIME}(g; f)$.

$\text{DTIME}(g) \subseteq \text{DTIME}(g; f)$ is trivial since $f(n) \geq n$ for all $n \in \mathbb{N}$.

Conversely let $L \in \text{DTIME}(g; f)$ and $\ell \in \mathbb{N}$ such that $a(\ell)$ is

$(g; f)$ -time-bounded deterministic TM that accepts L ; i.e. $L(a(\ell)) = L$.

Theorem of Borodin-Trakhtenbrot

$$t_\ell(m) = \sup\{n \mid w \in \Sigma^{\leq m-2}, (\kappa_0, \dots, \kappa_n) \text{ computation of } a(\ell) \text{ for } w\}$$
$$M_m = \{t_\ell(m) \mid \ell \leq m\}$$

Proof (3/3)

Exists $g(m) \in \mathbb{N}$ for every $m \geq 2$ such that $M_m \cap [g(m), f(g(m))] = \emptyset$.

It remains to prove $\text{DTIME}(g) = \text{DTIME}(g; f)$.

$\text{DTIME}(g) \subseteq \text{DTIME}(g; f)$ is trivial since $f(n) \geq n$ for all $n \in \mathbb{N}$.

Conversely let $L \in \text{DTIME}(g; f)$ and $\ell \in \mathbb{N}$ such that $a(\ell)$ is

$(g; f)$ -time-bounded deterministic TM that accepts L ; i.e. $L(a(\ell)) = L$.

Then $t_\ell(m) \leq f(g(m))$ for all $m \geq 2$ (by time bound) and additionally

$t_\ell(m) < g(m)$ for all $m \geq \ell$ (by property of g).

Theorem of Borodin-Trakhtenbrot

$$t_\ell(m) = \sup\{n \mid w \in \Sigma^{\leq m-2}, (\kappa_0, \dots, \kappa_n) \text{ computation of } a(\ell) \text{ for } w\}$$
$$M_m = \{t_\ell(m) \mid \ell \leq m\}$$

Proof (3/3)

Exists $g(m) \in \mathbb{N}$ for every $m \geq 2$ such that $M_m \cap [g(m), f(g(m))] = \emptyset$.

It remains to prove $\text{DTIME}(g) = \text{DTIME}(g; f)$.

$\text{DTIME}(g) \subseteq \text{DTIME}(g; f)$ is trivial since $f(n) \geq n$ for all $n \in \mathbb{N}$.

Conversely let $L \in \text{DTIME}(g; f)$ and $\ell \in \mathbb{N}$ such that $a(\ell)$ is

$(g; f)$ -time-bounded deterministic TM that accepts L ; i.e. $L(a(\ell)) = L$.

Then $t_\ell(m) \leq f(g(m))$ for all $m \geq 2$ (by time bound) and additionally $t_\ell(m) < g(m)$ for all $m \geq \ell$ (by property of g). Deterministic TM can simulate TM $a(\ell)$ and recognize finitely many inputs w of length smaller than $\ell - 2$ in at most $\ell \leq g(\ell)$ steps and directly switch into suitable final state. Hence $L \in \text{DTIME}(g)$. □

§2.3 Definition (Time- & space-constructible)

Let $f: \mathbb{N} \rightarrow \mathbb{N}$

- f is **time-constructible** if $f(n) \geq n$ for all $n \in \mathbb{N}$ and there exists det. TM that stops on input a^n after exactly $f(n+2)$ steps for all $n \in \mathbb{N}$

§2.3 Definition (Time- & space-constructible)

Let $f: \mathbb{N} \rightarrow \mathbb{N}$

- f is **time-constructible** if $f(n) \geq n$ for all $n \in \mathbb{N}$ and there exists det. TM that stops on input a^n after exactly $f(n+2)$ steps for all $n \in \mathbb{N}$
- f is **space-constructible** if $f(n) \geq \log n$ for all $n \in \mathbb{N}$ and there exists f -space-bounded det. TM that computes $a^n \mapsto a^{f(n+2)}$ for all $n \in \mathbb{N}$

§2.3 Definition (Time- & space-constructible)

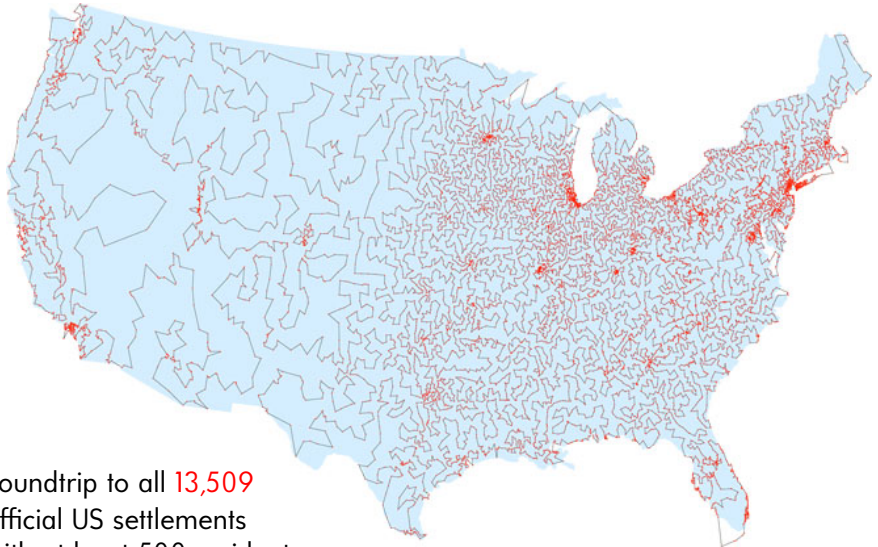
Let $f: \mathbb{N} \rightarrow \mathbb{N}$

- f is **time-constructible** if $f(n) \geq n$ for all $n \in \mathbb{N}$ and there exists det. TM that stops on input a^n after exactly $f(n+2)$ steps for all $n \in \mathbb{N}$
- f is **space-constructible** if $f(n) \geq \log n$ for all $n \in \mathbb{N}$ and there exists f -space-bounded det. TM that computes $a^n \mapsto a^{f(n+2)}$ for all $n \in \mathbb{N}$

Notes

- $\text{DTIME}(f)$ considered only for time-constructible f
- $\text{DSpace}(f)$ considered only for space-constructible f
- Function g in proof of Theorem of Borodin-Trakhtenbrot is not time-constructible in general

Traveling Salesman Problem



Roundtrip to all **13,509**
official US settlements
with at least 500 residents
(as of 1998)

(© David Applegate, Robert Bixby, Vasek Chvatal und William Cook)

Traveling Salesman Problem

Let $G = (E, K, g)$ finite weighted graph with $g: K \rightarrow \mathbb{N}_+$

Roundtrip W is cycle of length $|E|$

i.e. path $W = e_0 \rightarrow e_1 \rightarrow \cdots \rightarrow e_n$ with $e_0 = e_n$ and $n = |E|$

Traveling Salesman Problem

Let $G = (E, K, g)$ finite weighted graph with $g: K \rightarrow \mathbb{N}_+$

Roundtrip W is cycle of length $|E|$

i.e. path $W = e_0 \rightarrow e_1 \rightarrow \cdots \rightarrow e_n$ with $e_0 = e_n$ and $n = |E|$

Weight of path $W = e_0 \rightarrow e_1 \rightarrow \cdots \rightarrow e_n$ is

$$g(W) = \sum_{i=1}^n g((e_{i-1}, e_i))$$

Traveling Salesman Problem

Let $G = (E, K, g)$ finite weighted graph with $g: K \rightarrow \mathbb{N}_+$

Roundtrip W is cycle of length $|E|$

i.e. path $W = e_0 \rightarrow e_1 \rightarrow \cdots \rightarrow e_n$ with $e_0 = e_n$ and $n = |E|$

Weight of path $W = e_0 \rightarrow e_1 \rightarrow \cdots \rightarrow e_n$ is

$$g(W) = \sum_{i=1}^n g((e_{i-1}, e_i))$$

Decision problem TSP

- **Input:** Finite weighted graph $G = (E, K, g)$ and $\ell \geq 0$
- **Output:** Does there exist roundtrip with weight at most ℓ ?

Traveling Salesman Problem

We often only consider classic decision problems, but there are closely related interesting “problems”

Computation problem TSP-C

- **Input:** Finite weighted graph $G = (E, K, g)$ and $\ell \geq 0$
- **Output:** Roundtrip W with $g(W) \leq \ell$ if such roundtrip exists

Optimization problem TSP-O

- **Input:** Finite weighted graph $G = (E, K, g)$
- **Output:** Weight-minimal roundtrip W if roundtrip exists

Traveling Salesman Problem

However, efficient solution to classical problem often yields efficient solution of variants

We show

$$\text{TSP} \in \mathbf{P} \rightarrow \text{TSP-O} \in \mathbf{P}$$

Traveling Salesman Problem

However, efficient solution to classical problem often yields efficient solution of variants

We show

$$\text{TSP} \in \mathbf{P} \rightarrow \text{TSP-O} \in \mathbf{P}$$

- Let $G = (E, K, g)$ be input graph and
A polynomial-time algorithm for TSP

Traveling Salesman Problem

However, efficient solution to classical problem often yields efficient solution of variants

We show

$$\text{TSP} \in \mathbf{P} \rightarrow \text{TSP-O} \in \mathbf{P}$$

- Let $G = (E, K, g)$ be input graph and A polynomial-time algorithm for TSP
- First check whether some roundtrip exists

$$A(G, \hat{\ell}) \quad \text{with} \quad \hat{\ell} = |E| \cdot \max\{g(k) \mid k \in K\}$$

Roundtrip exists iff roundtrip with weight at most $\hat{\ell}$ exists

Traveling Salesman Problem

- Determine optimal weight ℓ^* via binary search

① Let $\underline{\ell} = 1$ and $\bar{\ell} = \hat{\ell}$

(trivial bounds)

Traveling Salesman Problem

- Determine optimal weight ℓ^* via binary search

① Let $\underline{\ell} = 1$ and $\bar{\ell} = \hat{\ell}$

(trivial bounds)

② While $\underline{\ell} < \bar{\ell}$ set $\ell \leftarrow \lceil \frac{\underline{\ell} + \bar{\ell}}{2} \rceil$ and

(average of bounds)

$$(\underline{\ell}, \bar{\ell}) \leftarrow \begin{cases} (\underline{\ell}, \ell) & \text{if } A(G, \ell) \\ (\ell + 1, \bar{\ell}) & \text{otherwise} \end{cases}$$

(upper or lower half)

Traveling Salesman Problem

- Determine optimal weight ℓ^* via binary search

① Let $\underline{\ell} = 1$ and $\bar{\ell} = \hat{\ell}$

(trivial bounds)

② While $\underline{\ell} < \bar{\ell}$ set $\ell \leftarrow \lceil \frac{\underline{\ell} + \bar{\ell}}{2} \rceil$ and

(average of bounds)

$$(\underline{\ell}, \bar{\ell}) \leftarrow \begin{cases} (\underline{\ell}, \ell) & \text{if } A(G, \ell) \\ (\ell + 1, \bar{\ell}) & \text{otherwise} \end{cases}$$

(upper or lower half)

③ Return $\ell^* = \underline{\ell}$

Traveling Salesman Problem

- Determine optimal weight ℓ^* via binary search

① Let $\underline{\ell} = 1$ and $\bar{\ell} = \hat{\ell}$ (trivial bounds)

② While $\underline{\ell} < \bar{\ell}$ set $\ell \leftarrow \lceil \frac{\underline{\ell} + \bar{\ell}}{2} \rceil$ and (average of bounds)

$(\underline{\ell}, \bar{\ell}) \leftarrow \begin{cases} (\underline{\ell}, \ell) & \text{if } A(G, \ell) \\ (\ell + 1, \bar{\ell}) & \text{otherwise} \end{cases}$ (upper or lower half)

③ Return $\ell^* = \underline{\ell}$



- Number of calls to A bounded by $(|K| \geq 1 \text{ and } |E| > 1)$

$$\log(\hat{\ell}) = \log |E| + \log \hat{g} \leq |G| \quad \text{with} \quad \hat{g} = \max\{g(k) \mid k \in K\}$$

$|G| = |E|^2 \lceil \log \hat{g} \rceil$ (weight matrix) or

$|G| = |K| \cdot (2 \lceil \log |E| \rceil + \lceil \log \hat{g} \rceil)$ (list of edges)

Traveling Salesman Problem

- Determine optimal weight ℓ^* via binary search

① Let $\underline{\ell} = 1$ and $\bar{\ell} = \hat{\ell}$ (trivial bounds)

② While $\underline{\ell} < \bar{\ell}$ set $\ell \leftarrow \lceil \frac{\underline{\ell} + \bar{\ell}}{2} \rceil$ and (average of bounds)

$(\underline{\ell}, \bar{\ell}) \leftarrow \begin{cases} (\underline{\ell}, \ell) & \text{if } A(G, \ell) \\ (\ell + 1, \bar{\ell}) & \text{otherwise} \end{cases}$ (upper or lower half)

③ Return $\ell^* = \underline{\ell}$



- Number of calls to A bounded by $(|K| \geq 1 \text{ and } |E| > 1)$

$$\log(\hat{\ell}) = \log |E| + \log \hat{g} \leq |G| \quad \text{with} \quad \hat{g} = \max\{g(k) \mid k \in K\}$$

$|G| = |E|^2 \lceil \log \hat{g} \rceil$ (weight matrix) or

$|G| = |K| \cdot (2 \lceil \log |E| \rceil + \lceil \log \hat{g} \rceil)$ (list of edges)

Traveling Salesman Problem

- Determine optimal weight ℓ^* via binary search

① Let $\underline{\ell} = 1$ and $\bar{\ell} = \hat{\ell}$ (trivial bounds)

② While $\underline{\ell} < \bar{\ell}$ set $\ell \leftarrow \lceil \frac{\underline{\ell} + \bar{\ell}}{2} \rceil$ and (average of bounds)

$(\underline{\ell}, \bar{\ell}) \leftarrow \begin{cases} (\underline{\ell}, \ell) & \text{if } A(G, \ell) \\ (\ell + 1, \bar{\ell}) & \text{otherwise} \end{cases}$ (upper or lower half)

③ Return $\ell^* = \underline{\ell}$



- Number of calls to A bounded by $(|K| \geq 1 \text{ and } |E| > 1)$

$$\log(\hat{\ell}) = \log |E| + \log \hat{g} \leq |G| \quad \text{with} \quad \hat{g} = \max\{g(k) \mid k \in K\}$$

$|G| = |E|^2 \lceil \log \hat{g} \rceil$ (weight matrix) or

$|G| = |K| \cdot (2 \lceil \log |E| \rceil + \lceil \log \hat{g} \rceil)$ (list of edges)

Traveling Salesman Problem

- Determine optimal weight ℓ^* via binary search

① Let $\underline{\ell} = 1$ and $\bar{\ell} = \hat{\ell}$ (trivial bounds)

② While $\underline{\ell} < \bar{\ell}$ set $\ell \leftarrow \lceil \frac{\underline{\ell} + \bar{\ell}}{2} \rceil$ and (average of bounds)

$(\underline{\ell}, \bar{\ell}) \leftarrow \begin{cases} (\underline{\ell}, \ell) & \text{if } A(G, \ell) \\ (\ell + 1, \bar{\ell}) & \text{otherwise} \end{cases}$ (upper or lower half)

③ Return $\ell^* = \underline{\ell}$



- Number of calls to A bounded by $(|K| \geq 1 \text{ and } |E| > 1)$

$$\log(\hat{\ell}) = \log |E| + \log \hat{g} \leq |G| \quad \text{with} \quad \hat{g} = \max\{g(k) \mid k \in K\}$$

$|G| = |E|^2 \lceil \log \hat{g} \rceil$ (weight matrix) or

$|G| = |K| \cdot (2 \lceil \log |E| \rceil + \lceil \log \hat{g} \rceil)$ (list of edges)

Traveling Salesman Problem

- Determine optimal weight ℓ^* via binary search

① Let $\underline{\ell} = 1$ and $\bar{\ell} = \hat{\ell}$

(trivial bounds)

② While $\underline{\ell} < \bar{\ell}$ set $\ell \leftarrow \lceil \frac{\underline{\ell} + \bar{\ell}}{2} \rceil$ and

(average of bounds)

$$(\underline{\ell}, \bar{\ell}) \leftarrow \begin{cases} (\underline{\ell}, \ell) & \text{if } A(G, \ell) \\ (\ell + 1, \bar{\ell}) & \text{otherwise} \end{cases}$$

(upper or lower half)

③ Return $\ell^* = \underline{\ell}$



- Number of calls to A bounded by

($|K| \geq 1$ and $|E| > 1$)

$$\log(\hat{\ell}) = \log |E| + \log \hat{g} \leq |G| \quad \text{with} \quad \hat{g} = \max\{g(k) \mid k \in K\}$$

$|G| = |E|^2 \lceil \log \hat{g} \rceil$ (weight matrix) or

$|G| = |K| \cdot (2 \lceil \log |E| \rceil + \lceil \log \hat{g} \rceil)$ (list of edges)

Traveling Salesman Problem

- Determine optimal weight ℓ^* via binary search

① Let $\underline{\ell} = 1$ and $\bar{\ell} = \hat{\ell}$ (trivial bounds)

② While $\underline{\ell} < \bar{\ell}$ set $\ell \leftarrow \lceil \frac{\underline{\ell} + \bar{\ell}}{2} \rceil$ and (average of bounds)

$(\underline{\ell}, \bar{\ell}) \leftarrow \begin{cases} (\underline{\ell}, \ell) & \text{if } A(G, \ell) \\ (\ell + 1, \bar{\ell}) & \text{otherwise} \end{cases}$ (upper or lower half)

③ Return $\ell^* = \underline{\ell}$



- Number of calls to A bounded by $(|K| \geq 1 \text{ and } |E| > 1)$

$$\log(\hat{\ell}) = \log |E| + \log \hat{g} \leq |G| \quad \text{with} \quad \hat{g} = \max\{g(k) \mid k \in K\}$$

$|G| = |E|^2 \lceil \log \hat{g} \rceil$ (weight matrix) or

$|G| = |K| \cdot (2 \lceil \log |E| \rceil + \lceil \log \hat{g} \rceil)$ (list of edges)

Traveling Salesman Problem

- Determine optimal weight ℓ^* via binary search

① Let $\underline{\ell} = 1$ and $\bar{\ell} = \hat{\ell}$ (trivial bounds)

② While $\underline{\ell} < \bar{\ell}$ set $\ell \leftarrow \lceil \frac{\underline{\ell} + \bar{\ell}}{2} \rceil$ and (average of bounds)

$(\underline{\ell}, \bar{\ell}) \leftarrow \begin{cases} (\underline{\ell}, \ell) & \text{if } A(G, \ell) \\ (\ell + 1, \bar{\ell}) & \text{otherwise} \end{cases}$ (upper or lower half)

③ Return $\ell^* = \underline{\ell}$



- Number of calls to A bounded by $(|K| \geq 1 \text{ and } |E| > 1)$

$$\log(\hat{\ell}) = \log |E| + \log \hat{g} \leq |G| \quad \text{with} \quad \hat{g} = \max\{g(k) \mid k \in K\}$$

$|G| = |E|^2 \lceil \log \hat{g} \rceil$ (weight matrix) or

$|G| = |K| \cdot (2 \lceil \log |E| \rceil + \lceil \log \hat{g} \rceil)$ (list of edges)

Traveling Salesman Problem

- Determine optimal weight ℓ^* via binary search

① Let $\underline{\ell} = 1$ and $\bar{\ell} = \hat{\ell}$

(trivial bounds)

② While $\underline{\ell} < \bar{\ell}$ set $\ell \leftarrow \lceil \frac{\underline{\ell} + \bar{\ell}}{2} \rceil$ and

(average of bounds)

$$(\underline{\ell}, \bar{\ell}) \leftarrow \begin{cases} (\underline{\ell}, \ell) & \text{if } A(G, \ell) \\ (\ell + 1, \bar{\ell}) & \text{otherwise} \end{cases}$$

(upper or lower half)

③ Return $\ell^* = \underline{\ell}$



- Number of calls to A bounded by

$$(|K| \geq 1 \text{ and } |E| > 1)$$

$$\log(\hat{\ell}) = \log |E| + \log \hat{g} \leq |G| \quad \text{with} \quad \hat{g} = \max\{g(k) \mid k \in K\}$$

$$|G| = |E|^2 \lceil \log \hat{g} \rceil \text{ (weight matrix) or}$$

$$|G| = |K| \cdot (2 \lceil \log |E| \rceil + \lceil \log \hat{g} \rceil) \text{ (list of edges)}$$

Traveling Salesman Problem

- Determine optimal weight ℓ^* via binary search

① Let $\underline{\ell} = 1$ and $\bar{\ell} = \hat{\ell}$ (trivial bounds)

② While $\underline{\ell} < \bar{\ell}$ set $\ell \leftarrow \lceil \frac{\underline{\ell} + \bar{\ell}}{2} \rceil$ and (average of bounds)

$(\underline{\ell}, \bar{\ell}) \leftarrow \begin{cases} (\underline{\ell}, \ell) & \text{if } A(G, \ell) \\ (\ell + 1, \bar{\ell}) & \text{otherwise} \end{cases}$ (upper or lower half)

③ Return $\ell^* = \underline{\ell}$



- Number of calls to A bounded by $(|K| \geq 1 \text{ and } |E| > 1)$

$$\log(\hat{\ell}) = \log |E| + \log \hat{g} \leq |G| \quad \text{with} \quad \hat{g} = \max\{g(k) \mid k \in K\}$$

$|G| = |E|^2 \lceil \log \hat{g} \rceil$ (weight matrix) or

$|G| = |K| \cdot (2 \lceil \log |E| \rceil + \lceil \log \hat{g} \rceil)$ (list of edges)

Traveling Salesman Problem

- Computation of optimal roundtrip (of weight ℓ^*) remains
 - ① Let $K_0 = K = \{k_1, \dots, k_m\}$ (enumeration of edges)

Traveling Salesman Problem

- Computation of optimal roundtrip (of weight ℓ^*) remains
 - 1 Let $K_0 = K = \{k_1, \dots, k_m\}$ (enumeration of edges)
 - 2 For all $i \in \{1, \dots, m\}$ in ascending order let (test if edge k_i contributes)

$$K_i \leftarrow \begin{cases} K_{i-1} \setminus \{k_i\} & \text{if } A((E, K_{i-1} \setminus \{k_i\}, g|_{K_{i-1} \setminus \{k_i\}}), \ell^*) \\ K_{i-1} & \text{otherwise} \end{cases}$$

$$g|_{K'} = g \cap (K' \times \mathbb{N}_+)$$

- 3 Return K_m

Traveling Salesman Problem

- Computation of optimal roundtrip (of weight ℓ^*) remains
 - 1 Let $K_0 = K = \{k_1, \dots, k_m\}$ (enumeration of edges)
 - 2 For all $i \in \{1, \dots, m\}$ in ascending order let (test if edge k_i contributes)

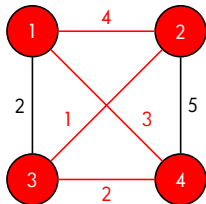
$$K_i \leftarrow \begin{cases} K_{i-1} \setminus \{k_i\} & \text{if } A((E, K_{i-1} \setminus \{k_i\}, g|_{K_{i-1} \setminus \{k_i\}}), \ell^*) \\ K_{i-1} & \text{otherwise} \end{cases}$$

$$g|_{K'} = g \cap (K' \times \mathbb{N}_+)$$

- 3 Return K_m
- Number of calls to A bounded by $|K| \leq |G|$

Traveling Salesman Problem

Graph with roundtrip of weight 10



Input graph
with opt. roundtrip



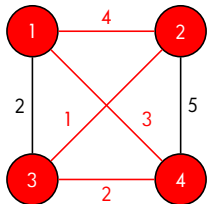
Removal edge (1, 3)



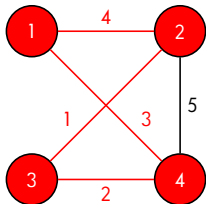
Removal edge (1, 2)
Length of roundtrip 11

Traveling Salesman Problem

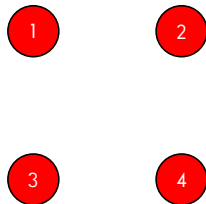
Graph with roundtrip of weight 10



Input graph
with opt. roundtrip



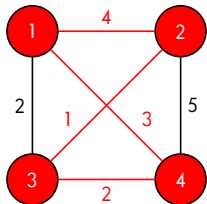
Removal edge (1, 3)



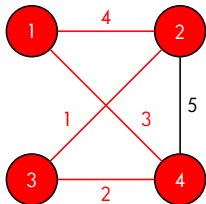
Removal edge (1, 2)
Length of roundtrip 11

Traveling Salesman Problem

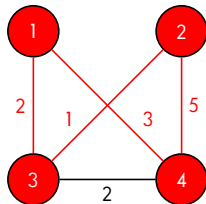
Graph with roundtrip of weight 10



Input graph
with opt. roundtrip



Removal edge (1, 3)



Removal edge (1, 2)
Length of roundtrip 11

Traveling Salesman Problem

Proof of correctness

For all $i \in \{0, \dots, m\}$:

- There exists roundtrip in $(E, K_i, g|_{K_i})$ with weight ℓ^*
- Every roundtrip W in $(E, K_i, g|_{K_i})$ with $g|_{K_i}(W) = \ell^*$ utilizes all edges of $\{k_1, \dots, k_i\} \cap K_i$

First statement is trivial by induction.

Traveling Salesman Problem

Proof of correctness

For all $i \in \{0, \dots, m\}$:

- There exists roundtrip in $(E, K_i, g|_{K_i})$ with weight ℓ^*
- Every roundtrip W in $(E, K_i, g|_{K_i})$ with $g|_{K_i}(W) = \ell^*$ utilizes all edges of $\{k_1, \dots, k_i\} \cap K_i$

First statement is trivial by induction. Suppose that there exists roundtrip W in $(E, K_i, g|_{K_i})$ with $g|_{K_i}(W) = \ell^*$ and edge $k_{i'} \in K_i$ with $i' \leq i$ such that $k_{i'}$ is not part of W .

Traveling Salesman Problem

Proof of correctness

For all $i \in \{0, \dots, m\}$:

- There exists roundtrip in $(E, K_i, g|_{K_i})$ with weight ℓ^*
- Every roundtrip W in $(E, K_i, g|_{K_i})$ with $g|_{K_i}(W) = \ell^*$ utilizes all edges of $\{k_1, \dots, k_i\} \cap K_i$

First statement is trivial by induction. Suppose that there exists roundtrip W in $(E, K_i, g|_{K_i})$ with $g|_{K_i}(W) = \ell^*$ and edge $k_{i'} \in K_i$ with $i' \leq i$ such that $k_{i'}$ is not part of W . Since $K_i \subseteq K_{i'-1}$ roundtrip W is roundtrip in $(E, K_{i'-1}, g|_{K_{i'-1}})$.

Traveling Salesman Problem

Proof of correctness

For all $i \in \{0, \dots, m\}$:

- There exists roundtrip in $(E, K_i, g|_{K_i})$ with weight ℓ^*
- Every roundtrip W in $(E, K_i, g|_{K_i})$ with $g|_{K_i}(W) = \ell^*$ utilizes all edges of $\{k_1, \dots, k_i\} \cap K_i$

First statement is trivial by induction. Suppose that there exists roundtrip W in $(E, K_i, g|_{K_i})$ with $g|_{K_i}(W) = \ell^*$ and edge $k_{i'} \in K_i$ with $i' \leq i$ such that $k_{i'}$ is not part of W . Since $K_i \subseteq K_{i'-1}$ roundtrip W is roundtrip in $(E, K_{i'-1}, g|_{K_{i'-1}})$. It is also roundtrip in $(E, K_{i'-1} \setminus \{k_{i'}\}, g|_{K_{i'-1} \setminus \{k_{i'}\}})$ since $k_{i'}$ is not part of W .

Traveling Salesman Problem

Proof of correctness

For all $i \in \{0, \dots, m\}$:

- There exists roundtrip in $(E, K_i, g|_{K_i})$ with weight ℓ^*
- Every roundtrip W in $(E, K_i, g|_{K_i})$ with $g|_{K_i}(W) = \ell^*$ utilizes all edges of $\{k_1, \dots, k_i\} \cap K_i$

First statement is trivial by induction. Suppose that there exists roundtrip W in $(E, K_i, g|_{K_i})$ with $g|_{K_i}(W) = \ell^*$ and edge $k_{i'} \in K_i$ with $i' \leq i$ such that $k_{i'}$ is not part of W . Since $K_i \subseteq K_{i'-1}$ roundtrip W is roundtrip in $(E, K_{i'-1}, g|_{K_{i'-1}})$. It is also roundtrip in $(E, K_{i'-1} \setminus \{k_{i'}\}, g|_{K_{i'-1} \setminus \{k_{i'}\}})$ since $k_{i'}$ is not part of W . Hence $k_{i'} \notin K_{i'}$ and $K_i \subseteq K_{i'}$ contradicting $k_{i'} \in K_i$. $\square$

Propositional Logic Satisfiability

Decision problem SAT

- **Input:** Propositional formula F
- **Output:** Is F satisfiable?

Propositional Logic Satisfiability

Decision problem SAT

- **Input:** Propositional formula F
- **Output:** Is F satisfiable?

Computation problem SAT-C

- **Input:** Propositional formula F
- **Output:** Model of F if one exists

Propositional Logic Satisfiability

Also in this case, efficient solution of classical decision problem yields efficient solution to computation problem

We show

$$\text{SAT} \in \mathbf{P} \rightarrow \text{SAT-C} \in \mathbf{P}$$

Propositional Logic Satisfiability

Also in this case, efficient solution of classical decision problem yields efficient solution to computation problem

We show

$$\text{SAT} \in \mathbf{P} \rightarrow \text{SAT-C} \in \mathbf{P}$$

- Let F input formula and A polynomial-time algorithm for SAT

Propositional Logic Satisfiability

Also in this case, efficient solution of classical decision problem yields efficient solution to computation problem

We show

$$\text{SAT} \in \mathbf{P} \rightarrow \text{SAT-C} \in \mathbf{P}$$

- Let F input formula and A polynomial-time algorithm for SAT
- Check whether F is satisfiable

$$A(F)$$

Propositional Logic Satisfiability

- Computation of model remains
- Let $\{v_1, \dots, v_k\}$ atoms that occur in F
 - 1 Set $F_0 = F$
 - 2 For all $i \in \{1, \dots, k\}$ in ascending order, let $F_i = F_{i-1}[v_i \leftarrow I(v_i)]$ with

$$I(v_i) = \begin{cases} 1 & \text{if } A(F_{i-1}[v_i \leftarrow 1]) \\ 0 & \text{otherwise} \end{cases} \quad (\text{check which variant is satisfiable})$$

Propositional Logic Satisfiability

- Computation of model remains
- Let $\{v_1, \dots, v_k\}$ atoms that occur in F
 - 1 Set $F_0 = F$
 - 2 For all $i \in \{1, \dots, k\}$ in ascending order, let $F_i = F_{i-1}[v_i \leftarrow I(v_i)]$ with

$$I(v_i) = \begin{cases} 1 & \text{if } A(F_{i-1}[v_i \leftarrow 1]) \\ 0 & \text{otherwise} \end{cases} \quad (\text{check which variant is satisfiable})$$

- 3 Return I

Propositional Logic Satisfiability

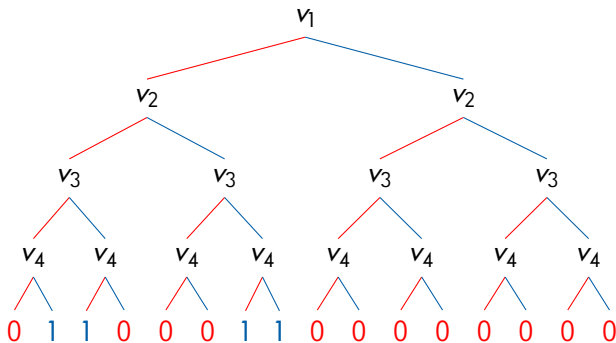
- Computation of model remains
- Let $\{v_1, \dots, v_k\}$ atoms that occur in F
 - 1 Set $F_0 = F$
 - 2 For all $i \in \{1, \dots, k\}$ in ascending order, let $F_i = F_{i-1}[v_i \leftarrow I(v_i)]$ with

$$I(v_i) = \begin{cases} 1 & \text{if } A(F_{i-1}[v_i \leftarrow 1]) \\ 0 & \text{otherwise} \end{cases} \quad (\text{check which variant is satisfiable})$$

- 3 Return I
- Number of calls to A bounded by

$$k \leq |F|$$

Variants of Problems



We stick to classical decision problems (languages)

Relations between Complexity Classes

§2.4 Theorem (Linear speed-up)

Let $L \in \text{DTIME}(f)$ with $f(n) \geq n$ for all $n \in \mathbb{N}$ and $\epsilon > 0$.

Then $L \in \text{DTIME}(g)$ with $g(n) = \lceil \epsilon f(n) \rceil + n$ for all $n \in \mathbb{N}$

Relations between Complexity Classes

§2.4 Theorem (Linear speed-up)

Let $L \in \text{DTIME}(f)$ with $f(n) \geq n$ for all $n \in \mathbb{N}$ and $\epsilon > 0$.

Then $L \in \text{DTIME}(g)$ with $g(n) = \lceil \epsilon f(n) \rceil + n$ for all $n \in \mathbb{N}$

Proof sketch

First compress input of length n by factor $m = \lceil \frac{12}{\epsilon} \rceil$ onto new tape.

This requires $n + 2 + \lceil \frac{n+2}{m} \rceil - 1$ steps including rewind.

Now execute TM M for L on compressed input and simulate m steps of M by just 6 steps (read neighboring cells in 4 steps and execute in 2 steps).

Relations between Complexity Classes

§2.4 Theorem (Linear speed-up)

Let $L \in \text{DTIME}(f)$ with $f(n) \geq n$ for all $n \in \mathbb{N}$ and $\epsilon > 0$.

Then $L \in \text{DTIME}(g)$ with $g(n) = \lceil \epsilon f(n) \rceil + n$ for all $n \in \mathbb{N}$

Proof sketch

First compress input of length n by factor $m = \lceil \frac{12}{\epsilon} \rceil$ onto new tape.

This requires $n + 2 + \lceil \frac{n+2}{m} \rceil - 1$ steps including rewind.

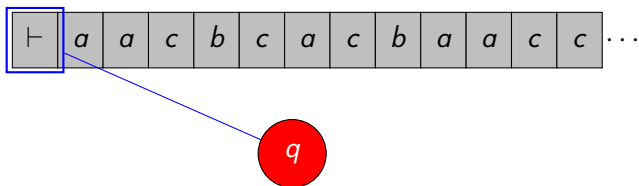
Now execute TM M for L on compressed input and simulate m steps of M by just 6 steps (read neighboring cells in 4 steps and execute in 2 steps).

Simulation requires $\leq \frac{6}{m}f(n+2) + \lceil \frac{n+2}{m} \rceil + n + 1$ steps and

$$\begin{aligned} \frac{6}{m}f(n+2) + \lceil \frac{n+2}{m} \rceil + n + 1 &\leq \frac{6\epsilon}{12}f(n+2) + \frac{\epsilon(n+2)}{12} + 1 + n + 1 \\ &= \frac{\epsilon}{2}f(n+2) + \underbrace{\frac{\epsilon}{2} \cdot \frac{n+2}{6}}_{\leq f(n+2)} + n + 2 \leq \lceil \epsilon f(n+2) \rceil + n + 2 = g(n+2) \quad \square \end{aligned}$$

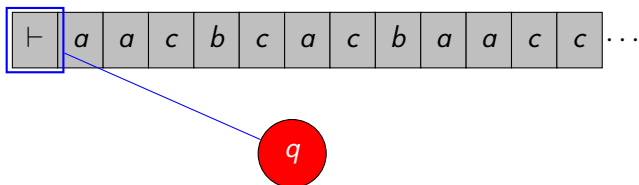
Relations between Complexity Classes

Input

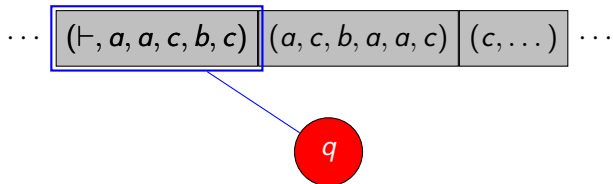


Relations between Complexity Classes

Input



Compressed Input



Relations between Complexity Classes

For function $f: \mathbb{N} \rightarrow \mathbb{N}$, let $\text{DTIME}(O(f)) = \bigcup_{g \in O(f)} \text{DTIME}(g)$
(we use same notation for DSPACE, NTIME, and NSPACE)

§2.5 Theorem (Linear tape compression)

$\text{DSPACE}(f) \subseteq \text{DSPACE}(g)$ for all $\epsilon > 0$ and $g(n) = \lceil \epsilon f(n) \rceil$ for all $n \in \mathbb{N}$

Relations between Complexity Classes

For function $f: \mathbb{N} \rightarrow \mathbb{N}$, let $\text{DTIME}(O(f)) = \bigcup_{g \in O(f)} \text{DTIME}(g)$
(we use same notation for DSPACE, NTIME, and NSPACE)

§2.5 Theorem (Linear tape compression)

$\text{DSPACE}(f) \subseteq \text{DSPACE}(g)$ for all $\epsilon > 0$ and $g(n) = \lceil \epsilon f(n) \rceil$ for all $n \in \mathbb{N}$

Consequences

- $\text{DSPACE}(O(f)) = \text{DSPACE}(f)$ and $\text{NSPACE}(O(f)) = \text{NSPACE}(f)$
- $\text{DTIME}(O(f)) = \text{DTIME}(f)$
if $\exists \epsilon > 0$ with $f(n) \geq (1 + \epsilon)n$ for all $n \in \mathbb{N}$
- $\text{NTIME}(O(f)) = \text{NTIME}(f)$
- (Multiplicative constants irrelevant)

Summary

- Theorem of Borodin-Trakhtenbrot & constructability
- Problems & their variants
- First relations between complexity classes

Second exercise sheet available in Moodle