

Poznámky z přednášek
Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

Algoritmy a datové struktury II

BONUS

Peter Černo, 2010
petercerno@gmail.com

Garant: prof. RNDr. Luděk Kučera, DrSc.

E-mail: Ludek.Kucera@mff.cuni.cz

Domácí stránka: <http://www.ms.mff.cuni.cz/acad/kam/kucera/>

Garant: doc. RNDr. Ondřej Čepek, Ph.D.

E-mail: Ondrej.Cepek@mff.cuni.cz

Domácí stránka: <http://ktiml.mff.cuni.cz/~cepek/>

Anotace: Pokračování přednášky Algoritmy a datové struktury I.

Syllabus:

1. Vyhledávání v textu:
algoritmy Aho-Corasicková, Knuth-Morris-Pratt, volitelně Rabin-Karp
2. Toky v sítích:
algoritmus zlepšující cesty (Ford-Fulkerson),
algoritmus Dinicův a Goldbergův,
volitelně hledání maximálního toku minimální ceny a párování v bi-partitním grafu
3. Algebraické algoritmy:
diskrétní Fourierova transformace, její motivace a aplikace,
algoritmus FFT (Fast Fourier Transform) a jeho implementace obvodem "butterfly"
volitelně příbuzné transformace (kosinová - komprese JPEG)
4. Paralelní aritmetické algoritmy:
třídící sítě (merge-sort nebo bitonic-sort)
sčítání čísel - algoritmus carry look-ahead
volitelně algoritmus Karatsuba-Ofman pro násobení čísel
5. Základní geometrické algoritmy v rovině:
konvexní obal
Voronoi diagram a Delanauy triangulace (algoritmus Fortune)

6. Převoditelnost problémů a třídy časové složitosti:
polynomiální transformace a redukce mezi rozhodovacími problémy
nedeterministické algoritmy, třídy P a NP
NP-úplnost
7. Aproximační algoritmy:
použití aproximačních algoritmů, poměrová a relativní chyba
jeden až dva příklady, např. knapsack, bin-packing, rozvrhování na
paralelních strojích,
včetně horního odhadu chyby
8. Pravděpodobnostní algoritmy a kryptografie:
algoritmy typu Monte Carlo (Rabin-Millerův test prvočíselnosti)
šifrování s otevřeným klíčem (RSA šifra)
volitelně DES a podobné algoritmy a další kryptografické protokoly
9. Dynamické programování

Literatura:

1. A.Koubková, J.Pavelka : Úvod do teoretické informatiky, Matfyzpress 1999
2. Aho, Hopcroft, Ullman : The design and analysis of computer algorithms, Addison-Wesley 1976
3. T.Cormen, Ch.Leiserson, R. Rivest, C. Stein : Introduction to Algorithms (2nd Edition), McGraw-Hill 2001
4. L.Kučera : Kombinatorické algoritmy, SNTL Praha 1983
5. L.Kučera, J. Nešetřil : Algebraické metody diskretní matematiky, SNTL Praha 1989
6. <http://kam.mff.cuni.cz/~ludek>

This page is intentionally left blank.

TINOG1 ALG. A DAT. STR. II - SKÚSKASYLABUS:

MHĽADÁVANIE VZORKOV V TEXTE

AHO-CORASICK : MHĽADAVACÍ STROJ, ALL. SPÄTNEJ FCIE

KNUTH-MORRIS-PRATT : KMP-MATCHER,

COMPUTE - PREFIX-FUNCTION

RABIN-KARP

TOKY V SIETVACH

FORD-FULKERSON

EDMONDS-KARP

DINIC

PUSH-RELABEL

RELABEL-TO-FRONT (GOLDBERG)

FAST FOURIER TRANSFORMATION

FFT

DISCRETE FFT

ITERATIVE FFT

SČÍTANIE

CARRY LOOKAHEAD

TRIEDNACIE SIETE

SORTING NETWORK

BITONIC SORTING NETWORK.

TRIEDY ROZHODOVACÍCH PROBLÉMOV

TURINGOV STROJ

SAT (CNF)

PROBLÉM KLUKY

NEZÁVISLÁ PŇNOŽINA

BATON

ROZDELENIE, SUBSET SUM.

KRYPTOGRAFIA

KOMUTUJÚCE ŠIFRY

VEREJNÝ KÚČ, HYBRIDNÉ ŠIF., AUTENTIZÁCIA,
CERTIFIKAČNÉ AUTORITY

ROZŠÍRENÝ EUKLIDOV ALG.

RSA (RIVEST SHAMIR ADELMAN)

PRAVEPODOBNOSTNÉ ALG.

TESTOVANIE PRVOČISELNOSTI

APROXIMAČNÉ ALGORITMY:

APROXIMAČNÁ SCHEMA

VRCHOLOVÉ POKRYTIE

OBCH. CESTOVÁCI

SÚČET PODMNOŽÍN

GEOMETRICKÉ ALGORITMY

KONVEXNÝ OBAL 2D

DYNAMICKÉ PROGRAMOVANIE

UHLOPRIEMKY V KONVEX. MNOHOKU.

LCS

MASTER THEOREM

TINOGI ALG. A DAT. STR. II - SKÚŠKAVYHLADÁVANIE VZORKOV V TEXTE

AHO - CORASICK

 Σ , ABECEDA, TEXT $x = \sigma_1 \dots \sigma_n \in \Sigma^*$ $K = \{y_1, y_2, \dots, y_k\} \subseteq \Sigma^*$ $y_i = \tau_{i,1} \dots \tau_{i,l(i)} \quad , \quad l = \sum_{i=1}^k l(i)$ CHCEME NÁJSŤ V DVOJICE $(i, y_j) : y_j \sqsupseteq \sigma_1 \dots \sigma_i$ AHO - CORASICK - MATCHER (f, g, out) state $\leftarrow 0$ for $i \leftarrow 1 \dots n$ dowhile $g(\text{state}, \sigma_i) = 1$ do
state $\leftarrow f(\text{state})$ state $\leftarrow g(\text{state}, \sigma_i)$ foreach $y \in \text{out}(\text{state})$ do
REPORT (i, y) ČASOVÁ ZLOŽITOSŤ
 $O(n)$

DŮKAZ SPRÁVNOSTI:

- A.
- KAŽDÝ STAV JEDN. KOREŠ. K PREDPONE NEJAKÉHO SLOVA Z K
 - NECH STATE REPR. RETAZEC v , POTOM $f(\text{STATE})$ JE STAV, KT. REPREZENTUJE NAJVÄČŠIU PRÍPONU, KT. JE ZÁROVEN PREDPONOU NEJAKÉHO RETAZCA Z K
 - $y \in \text{out}(\text{state}) \Rightarrow y$ JE PRÍPONA SLOVA REPR. STATE

- B.
- INVARIANT: PO PREČÍTANÍ ZNAKU σ_i SA ALG. NACHYŤA V STAVE, KT. REPR. NAJDHÚJU PRÍPONU $\sigma_1 \dots \sigma_i$ TAKÚ, KT. JE ZÁROVEN PREDPONOU NEJAKÉHO SLOVA Z K

AHO - CORASICK - FORWARD

```

for  $i \leftarrow 1 \dots k$  do
  INSERT( $\gamma_i$ )
foreach  $\sigma \in \Sigma$  do
  if  $g(0, \sigma) = \perp$  then
     $g(0, \sigma) \leftarrow 0$ 

```

INSERT($\pi_1 \pi_2 \dots \pi_j$)

```

state  $\leftarrow 0$ 
for  $i \leftarrow 1 \dots j$  do
  if  $g(\text{state}, \pi_i) = \perp$  then
    count  $\leftarrow$  count + 1
     $g(\text{state}, \pi_i) \leftarrow$  count
     $g(\text{count}, *) \leftarrow \perp$ 
  state  $\leftarrow g(\text{state}, \pi_i)$ 

```

AHO - CORASICK - BACKWARD

```

 $f(0) \leftarrow 0, Q \leftarrow \emptyset$ 
foreach  $\sigma \in \Sigma$  do
  if  $r \leftarrow g(0, \sigma) \neq \perp$  then
     $f(r) \leftarrow 0$ 
     $Q.\text{enqueue}(r)$ 
while  $Q \neq \emptyset$  do
   $s \leftarrow Q.\text{dequeue}$ 
  foreach  $\sigma \in \Sigma$  do
    if  $r \leftarrow g(s, \sigma) \neq \perp$  then
       $s' \leftarrow f(s)$ 
      while  $g(s', \sigma) = \perp$  do
         $s' \leftarrow f(s')$ 
       $f(r) = g(s, \sigma), \text{out}(r) \leftarrow \text{out}(r) \cup \text{out}(f(r))$ 
       $Q.\text{enqueue}(r)$ 

```

INVARIANT:

VO FRONTE Q SA NACH.
 MA STAN S VE URDENEN f

ČASOVÁ ZLOŽITOST

$O(e)$

TINOG1 ALG. A DAT. ŠTR. II - SKÚSKA

KNUTH - MORRIS - PRATT

MAJME 1 ROK $P[1..m]$ A TEXT $T[1..n]$
 PREFIXOVÁ FČIA. $\pi[i] = \max \{k < i, P_k = T_i\}$

KMP-MATCHER

```

state ← 0
for i ← 1 .. n do
  while state > 0 ∧ P[state+1] ≠ T[i] do
    state ← π[state]
  if P[state+1] = T[i] then
    state ← state + 1
  if state = m then
    REPORT i
    state ← π[state]

```

$O(n)$

KMP - PREFIX

```

π[1] ← 0
k ← 0
for i ← 2 .. m do
  while k > 0 ∧ P[k+1] ≠ P[i] do
    k ← π[k]
  if P[k+1] = P[i] then
    k ← k + 1
  π[i] ← k

```

$O(m)$

INVARIANT
 $k = \pi[i-1]$

RABIN-KARP

base $\leftarrow |\Sigma|$

patt $\leftarrow 0$, text $\leftarrow 0$

for $i \leftarrow 1 \dots m$ do

 patt $\leftarrow \text{base} * \text{patt} + P[i]$

 text $\leftarrow \text{base} * \text{text} + T[i]$

tmp $\leftarrow 1$

for $i \leftarrow 1 \dots m-1$ do

 tmp $\leftarrow \text{base} * \text{tmp}$

$i \leftarrow 0$

while $i < n-m$ do

 if text = patt then

 if $T[i+1 \dots i+m] = P[1 \dots m]$ then REPORT $i+m$

$i \leftarrow i+1$

 if $i < n-m$ then

 text $\leftarrow \text{base} * (\text{text} - \text{tmp} * T[i]) + T[i+m]$

$O((n-m+1)m)$

TINOB1 ALG. A DAT. STR. II - SKÚŠKA

TOKY V SIETVACH

$$|f| = f(s, V) = f(V, V) - f(V-s, V) =$$

$$= f(V, V-s) = f(V, V-s-t) + f(V, t) = f(V, t)$$

NECH (S, T) JE REZ, $\cap$. $S, T \subseteq V$, $S \cap T = \emptyset$, $S \cup T = V$,
 $s \in S$, $t \in T$. POTOM $f(S, T) = |f|$

$$f(S, T) = f(S, V) - f(S, S) = f(s, V) + f(S-s, V) = f(s, V) = |f|$$

$$|f| = f(S, T) \leq c(S, T)$$

VEĽA: NÁSLEDUJÚCE PODMIENKY SÚ EKVIVALENTNÉ:

- f JE MAXIMÁLNY TOK
- G_f NEODBAHŤE CESTU Z s DO t
- $|f| = c(S, T)$ PRE NEJAKÝ REZ (S, T)

DŮKAZ: a) $\Rightarrow$ b) JE ZREJME

b) $\Rightarrow$ c) NECH S JE MIN. VRCHOLOV TAKÝCH,
 ŽE EXISTUJE CESTA Z s DO x V G_f , $T := V - S$.

POTOM $s \in S$, $t \notin S \Rightarrow t \in T$, (S, T) JE REZ

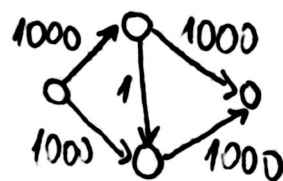
ZREJME $u \in S$, $v \in T \Rightarrow f(u, v) = c(u, v)$ $c_f(u, v) = 0$

$$\Rightarrow f(S, T) = c(S, T)$$

c) $\Rightarrow$ a) JE ZREJME

GENÉRIKÝ FORD-FULKERSON

CELOČÍSLNÉ KAPACITY $\Rightarrow O(|E||f^*|)$



EDMONDS-KARP

ROZŠIROVÚCU CESTU HĽADÁME PREHLADÁVANÍM DO ŠÍRKY

VETA1: VZDIALENOSŤ $\delta_f(s, v)$ JE POČAS VHK. E.-K. NEKLESAJÚCA

DOKAZ: (SPOROŤ) NECH v JE NAJBLIŽŠÍ VRCHOL OD s TAKÝ, ŽE
PO PRECHODE Z $f \rightarrow f'$ JE $\delta_{f'}(s, v) < \delta_f(s, v)$

NECH $s \rightsquigarrow u \rightarrow v$ JE CESTA $\Rightarrow v \in G_{f'}$

$$\delta_{f'}(s, v) = \delta_{f'}(s, u) + 1$$

$$\delta_{f'}(s, u) \geq \delta_f(s, u)$$

KEĎ $(u, v) \in E_f$, PODOŤ $\delta_f(s, v) \leq \delta_f(s, u) + 1 \leq \delta_{f'}(s, u) + 1 = \delta_{f'}(s, v)$ ∇

$\Rightarrow$ NUTNE $(u, v) \notin E_f$

VIDÍME, ŽE $(u, v) \notin E_f$, $(u, v) \in E_{f'}$ $\Rightarrow$ ~~NEKLESAJÚCA~~ DOĽ PRIDANÍ
TOK Z v DO $u \Rightarrow \delta_f(s, u) = \delta_f(s, v) + 1$ (NAJKR. CESTA DO u
IDE CEZ v)
 $\delta_f(s, v) = \delta_f(s, u) - 1 \leq \delta_{f'}(s, u) - 1 = \delta_{f'}(s, v) - 2$ ∇ SPOR.

VETA2: (u, v) JE KRITICKÁ NA p V G_f KEĎ

$$G(p) = G_f(u, v).$$

KAŽDÁ HRANA SA MOŽE STAŤ KRITICKOU NAJVIAC $\frac{|V|}{2}$ KRAŤ

DOKAZ: PREDP. ŽE SA HRANA (u, v) STALA KRITICKOU
NA NEAKKEJ CESTE p PO PRVÝ KRAŤ. $\delta_f(s, v) = \delta_f(s, u) + 1$

PO ROZŠÍRENÍ TOKU O f ZDŮRNE (u, v) Z G_f
NATO, ABY SA (u, v) STALA ZNOVU KRITICKOU,

MUSÍME ZUŠŤŤ TOK Z v DO u

$$\text{t.j. } \delta_{f'}(s, u) = \delta_{f'}(s, v) + 1 \geq \delta_f(s, v) + 1 = \delta_f(s, u) + 2$$

$\Rightarrow$ KEĎ SA (u, v) STANE KRITICKOU PO DRUHÝ KRAŤ,

$$\text{BUDE } \delta_{f'}(s, u) \geq \delta_f(s, u) + 2$$

KEĎŽE VZDIALENOSŤ VRCHOLU OD s JE OHR. ZHOREA $|V| - 2$

MOŽE SA HRANA STAŤ KRITICKOU NAJVIAC $\frac{|V|}{2}$ KRAŤ

$\Rightarrow$ PREHLADÁVANIE DO ŠÍRKY V $O(E)$ NA KA

ČASOVÚ ZLOŽITOSŤ $O(VE^2)$.

TINOG1 ALG. A DAT. STR II - KÚŠKA

DINIC

DISTANCE LABELS d BLOCKING FLOWS NÁJDE V ČASE $O(n \cdot m)$ ROZŠIROVACIU CESTU NÁJDE V ČASE $O(n)$ $d(s)$ MOŽE VZRAŠŤ n -KRAŤ $\Rightarrow$ ČASOVÁ ZLOŽITOSŤ $O(n^2 m)$

VYUŽÍVA DAT. STRUKTÚRU current-arc data structure

VÝPOČET DINICOVHO ALGORITMU MOŽE ROZDELÍŤ DO FÁZ

- o NA ZACIATKU KAŽDEJ FÁZY SPÓČÍTAME DISTANCE LABELS HRANA (u, v) JE PRÍPUSTNÁ (ADMISSIBLE) IBA AK

$$d(u) = d(v) + 1 \quad \text{A} \quad v \text{ NIE JE BLOKOVANÝ}$$

- o POČAS FÁZY NEMENÍME DISTANCE LABELS

- o V RÁMCI 1 FÁZY ~~ROZŠIROVACIU~~ NÁJDEME NAJVIAC m ROZŠÍRENÍ, PRETOŽE KAŽDE ROZŠÍRENIE SPÔSOBÍ, ŽE ASPOŇ 1 HRANA SA STANE NEPRÍPUSTNÁ (INADMISSIBLE)

- o FÁZA KONČÍ, KED SIET NEOBVIAHNE PRÍPUSTNÚ CESTU Z s DO t

PO PREPOČITANÍ DISTANCE LABELOV MUSÍ $d(s)$ NARAŠŤ

DINIC

NASTAV TOK f NA 0NOVÁ FÁZA: NASTAV DISTANCE LABELS $d(u)$ A OZNAČ VROCHTY AKO NEBLOKOVANÉ KED JE $d(s)$ DEFINOVANÉ POTOM ~~AKO NOVÁ FÁZA~~ $i \leftarrow s$ A ZAČ. FÁZU: $\rightarrow$ KED MAJ i PRÍPUSTNÚ HRANU, POTOMadvance(i)if $i = t$ then

augment

 $i \leftarrow s$ INAK ~~AKO NOVÁ FÁZA~~ if $i \neq s$ then retreat(i)
else PREJDI NA DALŠIU FÁZU

advance(i)

KEY DE (i,j) ADMISSIBLE, POUON

pred[j] $\leftarrow$ i, i $\leftarrow$ j

retreat(i)

blocked[i] $\leftarrow$ true

if i $\neq$ s then i $\leftarrow$ pred[i]

TINOG1 ALG. A DAT. STR. II - SKÚSKAPUSH-RELABEL ALGORITHM

PREFLOW $f(u,v) = -f(v,u)$, $f(u,v) \leq c(u,v)$, $f(v,u) = c(u)$ $u \neq s$
 VÝKONNÁ FUNKCIA (ANALOGIA DISTANCE LABELS)

$h: V \rightarrow \mathbb{N}_0$, $h(s) = |V|$, $h(t) = 0$, $\forall (u,v) \in E$ $h(u) \leq h(v) + 1$

HOVORÍME, ŽE VRCHOL $u \in V - \{s, t\}$ PRETEKÁ
 $\Leftrightarrow c(u) > 0$ (s ANI t NENÁLEŽU PRETEKÁM)

INITIALIZE-PREFLOW (G, s)

foreach $u \in V_G$ do
 $e[u] \leftarrow 0$, $h[u] \leftarrow 0$
 foreach $(u,v) \in E_G$ do
 $f[u,v] \leftarrow 0$, $f[v,u] \leftarrow 0$
 $h[s] \leftarrow |V_G|$

foreach $(s,u) \in E_G$ do
 $f[s,u] \leftarrow c(s,u)$
 $f[u,s] \leftarrow -f[s,u]$
 $e[s] \leftarrow e[s] - c(s,u)$
 $e[u] \leftarrow c(s,u)$

GENERIC-PUSH-RELABEL

INITIALIZE-PREFLOW (G, s)

while $\exists$ APLIKOVATEĽNÁ OP. PUSH ALEBO RELABEL do
 VYKONAJ TÚTO OPERÁCIU

PUSH (u,v)

IBA AK $h[u] = h[v] + 1$
 A u PRETEKÁ, T. J.
 $c[u] > 0$ $u \neq s, t$
 $d_s(u,v) \leftarrow \min(e[u], c(u,v))$
 $f[u,v] \leftarrow f[u,v] + d_s(u,v)$
 $f[v,u] \leftarrow -f[u,v]$
 $e[u] \leftarrow e[u] - d_s(u,v)$, $e[v] \leftarrow e[v] + d_s(u,v)$

RELABEL (u)

IBA AK u PRETEKÁ A
 $\forall (u,v) \in E$ $h[u] \leq h[v]$
 $h[u] \leftarrow 1 + \min_{(u,v) \in E} \{h[v]\}$

- AK $u \in V - \{s, t\}$ PRETEKA, POTOM MOŽNO APL. PUSH ALEBO RELABEL
- $h[u]$ NEKIESA'
- h JE STALE PÍSKOVÁ FUNKCIA
- NIEEXISTUJE CESTA $z : s \text{ DO } t \text{ V } G_z$
- PRE KAŽDÝ PRETEKAJÚCI VRCHOL u EXISTUJE $u \rightarrow s$

• KEĎ ŽOHNÝ
GENERIC-PUSH-RELABEL,
POTOM t JE MAX. TOK

$U = \{v, \text{EXISTUJE CESTA } z u \text{ DO } v \text{ V } G_z\},$
PRE SPOR NECH $s \notin U, \bar{U} = V - U. \quad \forall u \in \bar{U}, v \in U \text{ JE } f(u, v) \leq 0$
 $\Rightarrow c(U) = f(V, U) = f(\bar{U}, U) + f(U, U) \leq 0$

- STALE PLATÍ $h[u] \leq 2|V| - 1$
- POČET RELABEL OPERÁCIÍ JE NAJVIAC $(2|V| - 1)(|V| - 2) < 2|V|^2$
- POČET SATURATING PUSHOV JE NAJVIAC $2|V||E|$
- POČET NONSATURATING PUSHOV JE NAJVIAC $4|V|^2(|E| + |V|)$

PREDP., ŽE NAJVIAC SAT. PUSH (u, v) , T. $h[u] = h[v] - 1$. K TOMU, AKY
ZNOVU NAJVIAC TENTO SAT. PUSH, MUSÍ BYŤ PUSKNUTÝ TOK
 $z v \text{ DO } u \quad (h[u] = h[v] + 1)$ T. $h[v]$ MUSÍ NARÁŠŤ ASPOŇ O 2
 $\Rightarrow$ KEĎ NAJVIAC SAT. PUSH MEDZI u A v , POTOM IVA TO, AKY
NAJVIAC SAT. PUSH ZNOVU, MUSÍ BUŤ u ALEBO v KRAŠŤ ASPOŇ O 2
 $\Rightarrow$ MEDZI u A v JE NAJVIAC $2|V|$ SAT. PUSHOV

PRI NONSAT. PUSH. DEFINUJEME POTENCIÁL $\Phi = \sum_{v: c(v) > 0} h[v]$
NA ZAČATKU JE $\Phi = 0$. RELABEL / SAT. PUSH MOŽE Φ ZVÝŠIŤ NAJVIAC O $2|V|$
NONSAT. PUSH ZNÍŽI Φ ASPOŇ O 1

- CELKOVÝ ČAS SPOTREBOVANÝ NA RELABEL OP. JE $O(|V|E)$
NA KAŽDÝ VRCHOL MOŽE BYŤ RELABEL APL. NAJVIAC $2|V| - 1$ KEĎ
 $\Rightarrow$ ČASOVÁ ZL. JE $\leq \sum_{u \in V} (2|V| - 1) \deg(u) = O(|V|E)$

TINOG1 ALG. A DAT. STR. II - KÚŠKAPRÍPUSTNÁ (ADMISSIBLE) HRANA, PRÍPUSTNÁ SIET $G_{s,h}$

- $G_{s,h}$ JE DAG
- PUSH NEPRÍDAVA DO $G_{s,h}$ NOVEJ PRÍP. HRANY
- RELABEL(u) Z u BUDE VIESŤ PR. HR. $\odot$ ALE DO u NIE

RELABEL-TO-FRONT (G, s, t)INITIALIZE - PREFLOW (G, s) $L \leftarrow V - \{s, t\}$ IN ARBITRARY ORDERforeach $u \in L$ docurrent[u] $\leftarrow$ head[$N(u)$] $u \leftarrow$ head[L]while $u \neq \text{nil}$ doold-height $\leftarrow h[u]$ DISCHARGE(u)if $h[u] > \text{old-height}$ thenMOVE u TO FRONT OF L $u \leftarrow \text{next}[u]$

INVARIANT:

L JE TOP. ZOTRIEDENÝ
ZORNÝM VRCHOV $G_{s,h}$
+ KAŽDÝ VRCHOL PRED
 u JE NEPRETEKAJÚCI

DISCHARGE(u)while $e[u] > 0$ do $v \leftarrow \text{current}[u]$ if $v = \text{nil}$ thenRELABEL(u)current[u] $\leftarrow$ head[$N(u)$]else if $c(u,v) > 0$ a $h[u] = h[v] + 1$ thenPUSH(u,v)else current[u] $\leftarrow$ next-neighbor[u]

INVARIANT:

VŠETKY VRCHOLY, KT. SU V
ZORNÁME $N(u)$ PRED v
Majú HRANU (u,v)
NEPRÍPUSTNÚ

OZNAČME FAZU AKO ČAS MEDZI DVOMA RELABEL OP.

 $\Rightarrow$ MÁME $O(|V|^2)$. FAZ A KAŽDÁ POZOSTA'VAZ $O(M)$ VOLANÍ DISCHARGE $\Rightarrow$ DISCHARGE SA ZAVOLA $O(|V|^3)$ KRÁT, NONSATURATING
PUSH SPŮSOBÍ UKONČENIE DISCHARGE $\Rightarrow$ ICH POČET JE $O(|V|^3)$.

TINOGA ALG. A DAT. STR II - SKÚPKAFFT

ČASOVÁ DOMÉNA: FUNKCIA NAPRÚDCA ČAS NA AMPLITU
 FREKVENČNÁ DOMÉNA: VÁHY A FÁZY ADOUOVANE
 S FREKVENCIAMI

POLYNÓM $A(x)$:
 KOEFICIENTOVÁ REPRESENTÁCIA
 VRODOVÁ REPRESENTÁCIA

PRINCIPÁĽNÝ n -TÝ KOREŇ JEDNOTKY $\omega_n = e^{2\pi i/n}$

$\omega_n^0, \dots, \omega_n^{n-1}$ TVORIA GRUPO VZHĽADOM NA NÁSOBENIE
 MÁ TAKÚ ISTÚ STRUKTÚRU AKO $(\mathbb{Z}_n, +)$

CANCELLATION LEMMA: $\omega_{dn}^{dk} = \omega_n^k$

$$\Rightarrow \omega_n^{n/2} = \omega_2 = -1$$

KEĎ JE n PÁRNE, POTOM $\omega_n^k = -\omega_n^{k+n/2}$

$$\{(\omega_n^0)^2, \dots, (\omega_n^{n-1})^2\} = \{\omega_{n/2}^0, \dots, \omega_{n/2}^{n/2-1}\} \quad (\text{HALVING LEMMA})$$

$$\text{KEĎ } k \text{ NIEDELI } n, \text{ POTOM } \sum_{j=0}^{n-1} (\omega_n^k)^j = \frac{(\omega_n^k)^n - 1}{\omega_n^k - 1} = 0$$

$$\text{KEĎ } k \text{ DELÍ } n, \text{ JE } \sum_{j=0}^{n-1} (\omega_n^k)^j = n$$

(SUMMATION LEMMA)

NECH $a = (a_0, \dots, a_{n-1})$ n JE POCNINA 2

$y_k = \sum_{j=0}^{n-1} a_j \omega_n^{kj}$, $y = (y_0, \dots, y_{n-1})$ JE DISKRÉTNÁ

FOURIEROVÁ TRANSFORMÁCIA (DFT) $y = \text{DFT}_n(a)$

$$A(x) = a_0 + a_1 x + \dots + a_{n-1} x^{n-1}$$

$$A^{[0]}(x) = a_0 + a_2 x + \dots + a_{n-2} x^{\frac{n}{2}-1}$$

$$A^{[1]}(x) = a_1 + a_3 x + \dots + a_{n-1} x^{\frac{n}{2}-1}$$

$$\text{PLATÍ } A(x) = A^{[0]}(x^2) + x A^{[1]}(x^2)$$

RECURSIVE - FFT (a)

$n \leftarrow \text{length}[a]$

if $n=1$ then return a

$\omega_n \leftarrow e^{2\pi i/n}$

$\omega \leftarrow 1$

$a^{[0]} \leftarrow (a_0, a_2, \dots, a_{n-2})$

$a^{[1]} \leftarrow (a_1, a_3, \dots, a_{n-1})$

$y^{[0]} \leftarrow \text{RECURSIVE-FFT}(a^{[0]})$

$y^{[1]} \leftarrow \text{RECURSIVE-FFT}(a^{[1]})$

for $k \leftarrow 0 \dots \frac{n}{2} - 1$ do

$y_k \leftarrow y_k^{[0]} + \omega y_k^{[1]}$

$y_{k+\frac{n}{2}} \leftarrow y_k^{[0]} - \omega y_k^{[1]}$

$\omega \leftarrow \omega \cdot \omega_n$

return y

$$y_k^{[0]} = A^{[0]}(\omega_n^k) = A^{[0]}(\omega_n^{2k})$$

$$y_k^{[1]} = A^{[1]}(\omega_n^k) = A^{[1]}(\omega_n^{2k})$$

$$A(\omega_n^k) = A^{[0]}(\omega_n^{2k}) + \omega_n^k A^{[1]}(\omega_n^{2k})$$

$$\begin{aligned} A(\omega_n^{k+\frac{n}{2}}) &= A^{[0]}(\omega_n^{2k+n}) + \omega_n^{k+\frac{n}{2}} A^{[1]}(\omega_n^{2k+n}) = \\ &= A^{[0]}(\omega_n^{2k}) - \omega_n^k A^{[1]}(\omega_n^{2k}) \end{aligned}$$

$$T(n) = 2T\left(\frac{n}{2}\right) + \Theta(n) = \Theta(n \log n)$$

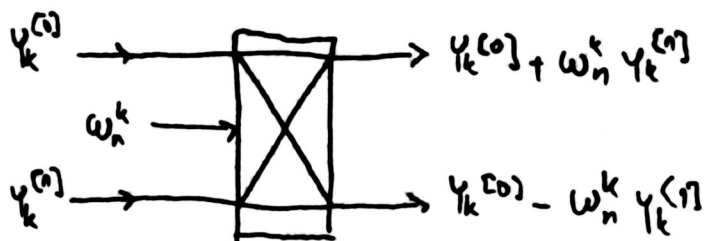
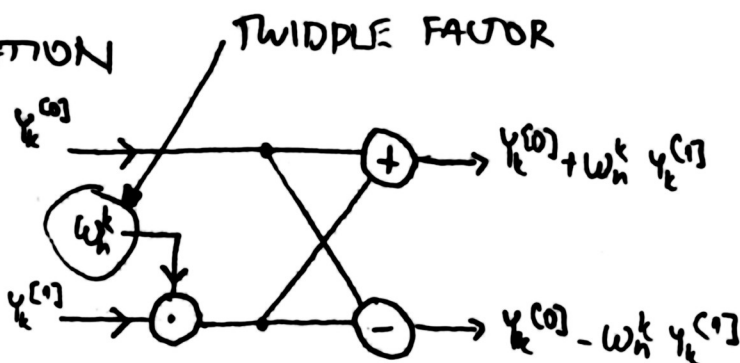
BUTTERFLY OPERATION

$t \leftarrow \omega y_k^{[1]}$

$y_k \leftarrow y_k^{[0]} + t$

$y_{k+\frac{n}{2}} \leftarrow y_k^{[0]} - t$

$\omega \leftarrow \omega \omega_n$



TINOGI ALG. A DAT. STR. II - SKÚSKA

ITERATIVE-FFT (GENERIC)

```

for s ← 1 to  $\log_2 n$  do
  for k ← 0 to n-1 by  $2^s$ 
    combine  $A[k \dots k+2^{s-1}-1]$  and  $A[k+2^{s-1} \dots k+2^s-1]$ 
    into  $A[k \dots k+2^s-1]$ 

```

ITERATIVE-FFT (a)

BIT-REVERSE-COPY (a, A)

 $n \leftarrow \text{length}[A]$

```

for s ← 1 to  $\log_2 n$  do

```

 $m \leftarrow 2^s$ $\omega_m \leftarrow e^{2\pi i/m}$

```

  for k ← 0 to n-1 by m do

```

 $w \leftarrow 1$

```

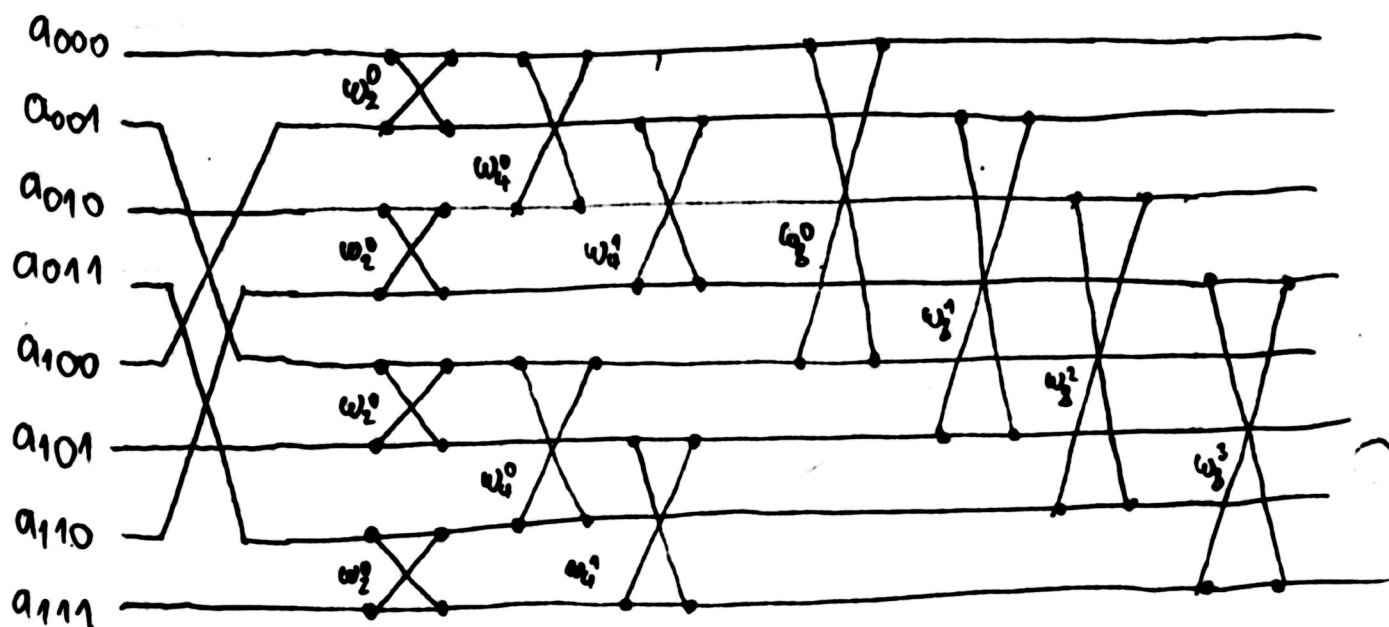
    for j ← 0 to  $\frac{m}{2} - 1$  do

```

 $t \leftarrow w A[k + \frac{m}{2} + j]$ $u \leftarrow A[k+j]$ $A[k+j] \leftarrow u+t$ $A[k + \frac{m}{2} + j] \leftarrow u-t$ $w \leftarrow \omega_m w$

BIT-REVERSAL PERMUTATION

PARALELNÝ FFT OBVOD



$$(ax+b)(cx+d) = \overset{\beta}{ac}x^2 + \overset{2-\beta-2}{(ad+bc)}x + \overset{2}{bd}$$

$$\begin{matrix} 2 \\ \beta \\ 2 \end{matrix} (a+b)(c+d) = ac + ad + bc + bd$$

$$T(n) = 3T\left(\frac{n}{2}\right) + \Theta(n)$$

$$T(n) = aT\left(\frac{n}{b}\right) + f(n)$$

$$\Rightarrow T(n) = \Theta(n^{\log_2 3})$$

$$a=3, b=2$$

$$f(n) = O(n^{\log_2 3 - \epsilon}) \quad \epsilon > 0$$

TINOGI ALG. A DAT. STR. II - SKÚŠKASORTING NETWORKS

HĺBKÁ SPOZU (DEPTH OF WIRE)



SORTING NETWORK JE COMPARISON NETWORK, PRE KTORÝ
VÝSTUPNÁ SEKVENCIA $b_1 \leq \dots \leq b_n$ PRE KAŽDÚ VSTUPNÚ
SEKVENCIU

KAŽDÁ SORTING NETWORK MÁ HĺBKU ASPOŇ $\log_2 n$
+ MÁ $\Omega(n \log_2 n)$ KOMPARETOROV

ZERO-ONE PRINCIPLE

LEMMA: KED COMP. NETW. TRANSFORMUJE

$a = (a_1, \dots, a_n)$ NA $b = (b_1, \dots, b_n)$,

f JE NEKLEŠAJÚCA FCA,

POTOM TÁTO COMP. NETW. TRANSFORMUJE

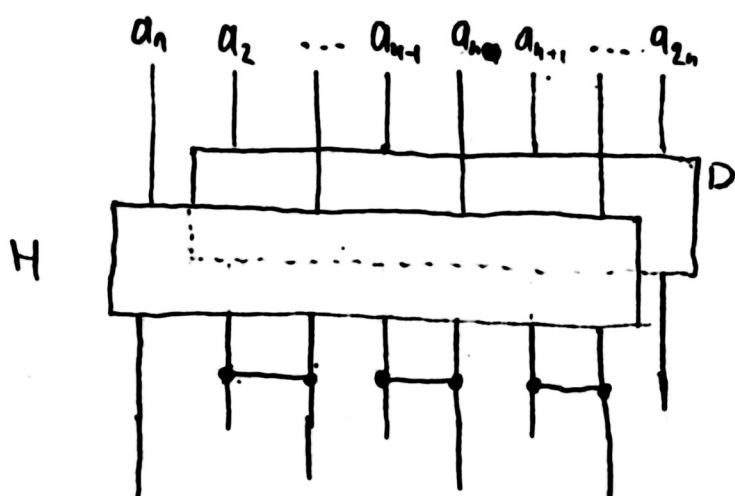
$f(a) = (f(a_1), \dots, f(a_n))$ NA $f(b) = (f(b_1), \dots, f(b_n))$

LEMMA (ZERO-ONE PRINCIPLE): KED COMP. NETW.
UTRIEDI VŠETKYCH 2^n SEKVENCÍ 0 A 1 SPRÁVNE,
POTOM JE TO SORTING NETWORK.

PRE SPOR NECH UTRIEDI V 2^n SEKV. , ALE $\exists a = (a_1, \dots, a_n)$:
 $a_i < a_j$, ALE SORT. NETW. POLOŽÍ a_j PRED a_i .

DEFINUJME $f(x) = \begin{cases} 0 & x \leq a_i \\ 1 & x > a_i \end{cases}$. KED APLIKUJEME SORT. NETW.

NA $f(a)$, BUDE $f(a_j)$ PRED $f(a_i)$, TUDŽ JE SPOR f .



$$a_1 \leq a_2 \leq \dots \leq a_n \quad a_{n+1} \leq \dots \leq a_{2n}$$

H	$a_1 \leq a_3 \leq \dots \leq a_{n-1}$	$a_{n+1} \leq a_{n+3} \leq \dots \leq a_{2n-1}$
D	$a_2 \leq a_4 \leq \dots \leq a_n$	$a_{n+2} \leq a_{n+4} \leq \dots \leq a_{2n}$

$$H \begin{smallmatrix} HD \\ DH \end{smallmatrix} < \begin{smallmatrix} HD \\ DH \end{smallmatrix} \dots D$$

TINOĎI ALG. A DAT. STR. II - SKÚŠKA

ARITMETICKÉ OBVODY

ÚLOHA: CHCEME SČÍTAT $\sum_{i=0}^{n-1} u_i 2^i$ A $\sum_{i=0}^{n-1} v_i 2^i$ $u_i, v_i \in \{0,1\}$
 $\rightarrow$ VÝSLEDOK JE $\sum_{i=0}^n s_i 2^i$

NAJVNÍ PRÍSTUP:

$$c_0 \leftarrow 0$$

$$s_i \leftarrow u_i \oplus v_i \oplus c_{i-1}$$

$$i = 0 \dots n-1$$

$$c_i \leftarrow \text{majority}(u_i, v_i, c_{i-1})$$

$$s_n \leftarrow c_{n-1}$$

HLĚBKŮ OBVODU JE $O(n)$

CARRY LOOKAHEAD:

FORMULNÉ ZAVEDENIE

$$p_i = u_i \oplus v_i \quad q_i = u_i \wedge v_i \quad i = 0 \dots n-1$$

$$\text{POŤON ZREJNÉ} \quad s_i = p_i \oplus c_{i-1}$$

$$c_i = q_i \vee (p_i \wedge c_{i-1})$$

DEFINOVANIE OPERATOR Δ

$$(g_1, p_1) \Delta (g_2, p_2) = (g_1 \vee (p_1 \wedge g_2), p_1 \wedge p_2)$$

 Δ JE ASSOCIATÍVNY:

$$\begin{aligned} ((g_1, p_1) \Delta (g_2, p_2)) \Delta (g_3, p_3) &= (g_1 \vee (p_1 \wedge g_2), p_1 \wedge p_2) \Delta (g_3, p_3) = \\ &= (g_1 \vee (p_1 \wedge g_2) \vee (p_1 \wedge p_2) \wedge g_3, (p_1 \wedge p_2) \wedge p_3) \\ (g_1, p_1) \Delta ((g_2, p_2) \Delta (g_3, p_3)) &= (g_1, p_1) \Delta (g_2 \vee (p_2 \wedge g_3), p_2 \wedge p_3) = \\ &= (g_1 \vee p_1 \wedge (g_2 \vee (p_2 \wedge g_3)), p_1 \wedge (p_2 \wedge p_3)) \quad \blacksquare \end{aligned}$$

$$\text{POLOŽME} \quad c_0 := 0 \quad G_0 := g_0 = u_0 \wedge v_0, \quad P_0 := p_0 = u_0 \oplus v_0$$

$$(G_i, P_i) = (g_i, p_i) \Delta (G_{i-1}, P_{i-1})$$

$$\text{POŤON} \quad c_i = G_i. \quad \text{DOKAZ:} \quad c_0 = g_0 \vee (p_0 \wedge c_{-1}) = g_0 = G_0$$

$$(G_i, P_i) = (g_i, p_i) \Delta (G_{i-1}, P_{i-1}) = (g_i \vee (p_i \wedge G_{i-1}), p_i \wedge P_{i-1}) = (c_i, P_i) \quad \blacksquare$$

$$P_i = p_i \wedge p_{i-1} \wedge \dots \wedge p_0 = (u_i \oplus v_i) \wedge \dots \wedge (u_0 \oplus v_0)$$

NP - ÚPLNOST

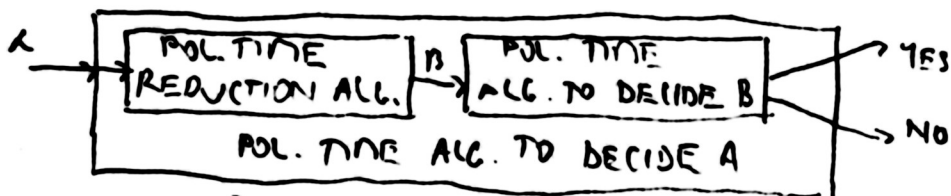
ZNAMY TURINGOV HALTING PROBLEM JE POČÍTAČOVĚ NERIEŠITELNÝ

P .. PROBLÉMY RIEŠITELNÉ V ČASE $O(n^k)$ PRE NEJAKÉ k
NP .. PROBLÉMY, KTORÉ SÚ VERIFIKOVATELNÉ V POL. ČASE
ZREJME $P \subseteq NP$

NPC .. JE NP, KED MÁ POL. RIEŠ. $\Rightarrow$ KNPC MÁV POL. RIEŠ.

VŤAH DECISION PROBLEMS VS OPTIMIZATION PROBLEMS

POLYNOMIAL TIME REDUCTION ALGORITHM



JAZYK L NAD ABECEDOU Σ JE AKÁKOLIEK MNOŽINA
REKURZÍV ZO Σ , $L \subseteq \Sigma^*$

PRE AKÁKOLIEK DECISION PROBLEM Q VIENE Q ZOBRAZIť
AKO JAZYK $L = \{x \in \Sigma^* : Q(x) = 1\}$ NAD $\Sigma = \{0, 1\}$

NAPR. PATH

PATH = $\{ \langle G, u, v, k \rangle : G = (V, E) \text{ JE NEORIENT. GRAF, } u, v \in V, \\ k \in \mathbb{N}_0, \text{ EXISTUJE CESTA Z } u \text{ DO } v \text{ POZOSTÁVAJÚCA Z MAX. } k \text{ H.} \}$

SAT $\in$ NPC
CNF-SAT $\in$ NPC
} CNF-SAT $\in$ NPC

TINOG1 ALG. A DAT. STR. II - SKUŠKA

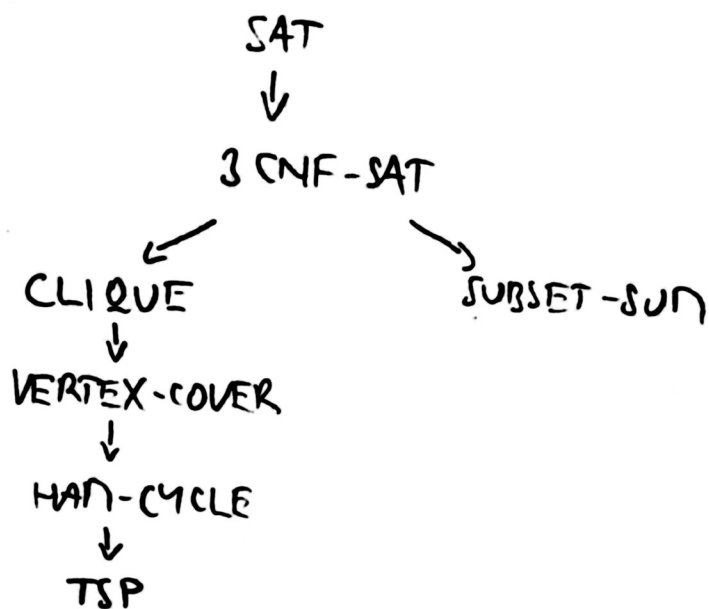
CLIQUE = $\{ \langle G, k \rangle : G \text{ JE GRAF S KLIKOU VEĽKOSTI } k \}$

VERTEX-COVER = $\{ \langle G, k \rangle : \text{GRAF } G \text{ MA' VRCHOLOVÉ POKRYTIE VEĽKOSTI } k \}$

HAM-CYCLE = $\{ \langle G \rangle : \text{V } G \text{ EXISTUJE CYKLUS ČEZ V VRCHOLOV} \}$

TSP = $\{ \langle G, c, k \rangle : G = (V, E) \text{ JE ÚPLNÝ GRAF, } c: V \times V \rightarrow \mathbb{Z}, k \in \mathbb{Z}, G \text{ MA' HAM CYKLUS CENY NAJVIAC } k \}$

SUBSET-SUM = $\{ \langle S, t \rangle : \text{EXISTUJE } S' \subseteq S : \sum_{x \in S'} x = t \}$



TINOG1 ALG. A DAT. STR. II - SKÚŠKA

RANDOMIZOVANÉ ALGORITMY
 PRAVDEPODOBNOSTNÉ ALGORITMY
 APROXIMÁČNÉ ALGORITMY
 HEURISTIKY

APROXIM. ALG.

PREDPOKLADAJE, ŽE MÁME OPTIMALIZAČNÝ PROBLÉM,
 KDE KAŽDÉ POTENCIÁLNE RIEŠENIE MÁ KLADNÚ CENU
 (MÔŽE SA JEDNAŤ O MAXIMALIZAČNÝ / MINIMALIZAČNÝ PROB.)

PONEROVA ČÍSLA (APPROXIMATION RATIO)

$$\rho(n) \geq \max\left(\frac{c}{c^*}, \frac{c^*}{c}\right)$$

c ... CENA RIEŠ. VÝPR. APROX. ALG.

c^* ... OPTIMÁLNÁ CENA

1-APROXIMÁČNÝ ALGORITMUS PRODUKUE VŽDY OPTIM. RIEŠENIE

APROXIMÁČNÁ SCHEMA PRE OPTIM. PROBLÉM JE
 APROX. ALGORITMUS, KT. AKO VSTUP BERIE INŠTANCIU
 PROBLÉMU A $\epsilon > 0$. JEDNÁ SA O $(1+\epsilon)$ -APROX. ALG.

POLYNOMIAL-TIME APPROX. SCHEME - PRE AKEĽNE $\epsilon > 0$
 BEŽY SCHEMA V POLYN. ČASE IZHŔADON NA n

FULLY POLYNOMIAL-TIME APPROX. SCHEME - ČASOVÁ
 ZLOŽITOSŤ JE POLYNOMIÁLNA VZHŔADON NA $\frac{1}{\epsilon}$ A n .

APPROX-VERTEX-COVER(G)

$$\sigma(n) = 2$$

$C \leftarrow \emptyset, E' \leftarrow E_G$

while $E' \neq \emptyset$ do

$(u,v) \in E', C \leftarrow C \cup \{u,v\},$

odstráni z E' hrany incidentné s u, v

return C

JE TO POLYNOMIAL-TIME
2-APPROX. ALG.

KEĎ OLN. A HRANY (u,v) VBERANÉ Z E' , POTOM ZREJME $|C'| \geq |A|$
 $|C| = 2|A| \Rightarrow |C| \leq 2|C'|$ ■

MN. A JE MAXIMAL MATCHING (POZOR! NIE MAXIMUM)

TSP (TRAVELING-SALESMAN PROBLEM)

Máme daný úplný neorientovaný graf $G = (V, E)$

s nezáporné ohodnotenými hranami c

keď c spĺňa Δ ineq $c(u,v) \leq c(u,w) + c(w,v)$, potom

EXISTUJE 2-APPROX. ALG. POLYN.-TIME

APPROX-TSP-TOUR(G, c)

NAJDI MINIMÁLNU KOSTRU T V G

$L \leftarrow$ ZOBRAZ Vrcholov v poradí PREORDER V T

return L

KEĎ VINECHÁME PREDPOKLAD O Δ -NEROVNOSTI, POTOM
NEEXISTUJE POLYN.-TIME APPROX. ALG. (KEĎ $P \neq NP$)

DŮKAZ: KEĎ EX. POL.-TIME σ -APPROX. ALG., POTOM

VIEME REDUKOVAŤ HAN-CYCLE NA TSP

$c(u,v) = \begin{cases} 1 & (u,v) \in E \\ \sigma M + 1 & \text{otherwise} \end{cases}$ ZÚŠŤOK JE ZREJMÝ. ■

SET COVERING PROBLEM

$(X, \mathcal{F})$ $\mathcal{F} \subseteq \mathcal{P}(X), \bigcup \mathcal{F} = X$

ÚLOHOU JE NAJÍT MIN. $\mathcal{C} \subseteq \mathcal{F} : \bigcup \mathcal{C} = X$

GREEDY METÓDA: V KAŽDOM KROKU VBERÁME $S \in \mathcal{F}$ TAKO,
KT. POKRYVA ČO NOVÝCH DOTERAZ NEPOKRYTÝCH VRCHOLOV

JE TO POLYNOMIÁLNY $\sigma(n)$ -APPROX. ALG., KDE

$$\sigma(n) = H(\max \{|S| : S \in \mathcal{F}\}) \quad H(n) = 1 + \frac{1}{2} + \dots + \frac{1}{n}$$

$$\sigma(n) \leq \ln(|X|) + 1$$

TINOG1 ALG. A DAT. STR. II - SKÚSKA

SUBSET-SUM PROBLEM

EXISTENCE EXPONENC. EXAKTNE' RIEŠENIE
A) FULLY POLYN.-TIME APPROX. SCHEMEEXACT-SUBSET-SUM (S, t) $n \leftarrow |S|$ $L_0 \leftarrow \langle 0 \rangle$ for $i \leftarrow 1$ to n do $L_i \leftarrow \text{MERGE-LISTS}(L_{i-1}, L_{i-1} + x_i)$ REMOVE FROM L_i ELEMENTS $> t$ return THE LARGEST ELEM. IN L

FULLY POLYNOMIAL-TIME APPROX. SCHEME

TRIMMING ZODKANDV L_i TJ. PRE DANE' $0 < \delta < 1$ ODSTRÁNIŤ Z L

ZO NAPVAC PRVKOV TAK, AŽ PRE KAŽDÝ

ODSTRÁŇENÝ $y \in L$ EXISTOVALO $z \in L'$ TAKÉ, ŽE

$$\frac{y}{1+\delta} \leq z \leq y$$

 z REPRESENTS y TRIM(L, δ) $L = \langle y_1, \dots, y_m \rangle$ $m \leftarrow |L|$ $L' \leftarrow \langle y_1 \rangle$, last $\leftarrow y_1$ for $i \leftarrow 2$ to m doif $y_i > \text{last} \cdot (1 + \delta)$ thenappend y_i to L' last $\leftarrow y_i$ return L' APPROX-SUBSET-SUM (S, t, ϵ)

SA LIST OD EXACT-SUBSET-SUM IDA O 1 RIADOK

PO MERGOVANÍ TREBA VIKONAT $L_i \leftarrow \text{TRIM}(L_i, \frac{\epsilon}{2^n})$ VRACA HODNOTU z , KT. SA OD OPT. z^* LIST VO FAKTOR $1 \pm \epsilon$

$$\frac{z^*}{1+\epsilon} \leq z \leq z^*$$

ΔΟΚΙΜΩΝΕ, ΩΤΙ APPROX-SUBSET-SUM ΕΙΝΑΙ FULLY POLYNOMIAL TIME APPROXIMATION SCHEME.

1. ΚΑΘΕ $y \in L_i$ ΠΑΡΤΙ ΜΕΣ ΔΕ $P_i \Rightarrow z^* \in L_n$ ΜΑΘ. ΩΤΙ $\leq y^* \in P_n$ ΜΑΘ.
2. ΙΝΔΥΚΤΙΟΥ ΔΟΚΙΜΩΝΕ, ΩΤΙ ΑΝ $y \in P_i$, ΤΟΤΟΤ $\exists z \in L_i$:

$$\frac{y}{(1+\epsilon/2n)^i} \leq z \leq y$$

ΓΕΝ $i=1$ ΠΑΤΙ, ΠΕΤΟΤΕ $L_1 = \text{TRIM}(P_1, \frac{\epsilon}{2n})$

ΥΠΟΘΕΤΕ $i > 1$. $y \in P_i$. Δ) ΚΕΔ $y \in P_{i-1}$, ΤΟΤΟΤ ΤΟ ΠΑΤΙ...

β) ΚΕΔ $y = x_i + y'$, $y' \in P_{i-1}$, ΤΟΤΟΤ $\exists z' \in L_{i-1}$:

$$\frac{y'}{(1+\epsilon/2n)^{i-1}} \leq z' \leq y'$$

$$\text{ΤΑΚΙΣΤΟ } \exists z \in L_i : \frac{z' + x_i}{1 + \frac{\epsilon}{2n}} \leq z \leq z' + x_i$$

$$\text{ΖΗΤΩΝΤΕ } z \leq z' + x_i \leq y' + x_i = y$$

$$\frac{z' + x_i}{1 + \epsilon/2n} \geq \frac{\frac{y'}{(1+\epsilon/2n)^{i-1}} + x_i}{1 + \epsilon/2n} = \frac{y'}{(1+\epsilon/2n)^i} + \frac{x_i}{1 + \epsilon/2n} \geq \frac{y' + x_i}{(1+\epsilon/2n)^i} = \frac{y}{(1+\epsilon/2n)^i}$$

$$\Rightarrow \frac{y}{(1+\epsilon/2n)^i} \leq z \leq y. \blacksquare$$

$$\Rightarrow \text{ΓΕΝ OPT. ΜΕΣ. } y^* \in P_n \exists z \in L_n : \frac{y^*}{(1+\frac{\epsilon}{2n})^n} \leq z \leq y^*$$

$$\Rightarrow \frac{y^*}{2} \leq (1+\frac{\epsilon}{2n})^n. \text{ ΚΕΔΩΤΕ } z^* \geq z, \text{ ΜΑΤΩΤΕ}$$

$$1 \leq y^*/z^* \leq (1+\frac{\epsilon}{2n})^n \leq e^{\epsilon/2} \leq 1 + \frac{\epsilon}{2} + (\frac{\epsilon}{2})^2 \leq 1 + \epsilon \blacksquare$$

ΕΙΣΤΕ ΔΟΚΙΜΩΝΕ, ΩΤΙ L_i ΜΑ' ΠΟΛ. ΔΕΔΟΥ

ΠΟ ΤΡΙΠΛΟΥΑΝΑ' ΔΥΑ ΤΟ ΣΕΒΕ ΙΔΙΟΤΕ $z, z' \in L_i$

ΜΑΤΑ ΣΤΕΝΑΤ $z'/z > 1 + \frac{\epsilon}{2n}$

$$\Rightarrow L_i \text{ ΜΑΤΕ ΠΑΤ ΜΑΤΩΤΕ } 2 + \lfloor \log_{1+\epsilon/2n} t \rfloor \text{ ΔΕΝΟΥ}$$

$$\begin{aligned} \log_{1+\epsilon/2n} t + 2 &\leq \frac{\ln t}{\ln(1+\epsilon/2n)} + 2 \leq \frac{\ln t}{\frac{\epsilon/2n}{1+\epsilon/2n}} + 2 = \frac{2n(1+\epsilon/2n) \ln t}{\epsilon} + 2 \\ &\leq \frac{4n \ln t}{\epsilon} + 2 \blacksquare \end{aligned}$$

TINOGA ALG. A DAT. STR. II - SKÚŠKAPROTOKOL POSIELANIA SPRÁV $A \rightarrow B$ CHCEME POSLAŤ SPRÁVU M OD A DO B

- A:
1. ZÍSKAŤE P_B (PUBLIC KEY)
 2. VIGENERUJEME K (SYMETRICKÚ ŠIFRU)
 3. NECH h JE HASH. FNCIA (MD5...), SPÓČÍTAME $h(M)$
 4. SPRÁVU ZASIFRUJEME POMOČOU $K \rightarrow K(M)$
 5. A PODPÍŠEME $S_A(h(M))$
 6. ZASIFRUJEME K POMOČOU $P_B \rightarrow P_B(K)$
 7. $\rightarrow B$ POSIELAME $(A, K(M), P_B(K), S_A(h(M)))$

HYBRIDNÉ
ŠIFROVANIE
HYBRIDNÁ
AUTENTIZÁCIA

- B:
1. DOŠTANEME SPRÁVU $(A, K(M), P_B(K), S_A(h(M)))$
 2. ZÍSKAŤE P_A
 3. DESIFRUJEME $K = S_B(P_B(K))$
 4. DESIFRUJEME $M = K(K(M))$
 5. SPÓČÍTAME $h(M)$ A POROVNAJEME S $P_A(S_A(h(M)))$

CERTIFIKAČNÉ AUTORITY

JE DANA AUTORITA Z CERTIFIKAŤ JE $(C, S_Z(C))$ C MOŽE BYŤ NAPR. = "ALICA MÁ VEREJNÝ KľÚČ P_A "

RSA (RIVEST, SHAMIR, ADELMAN)

1. ZVOĽTE DVE VEĽKÉ PRVOCÍSLA p, q
2. POLOŽTE $n = pq$, $\varphi = \varphi(n) = (p-1)(q-1)$
3. ZVOĽTE e NESÚDELNÉLNÉ r A NÁJDITE d : $ed \equiv 1 \pmod{\varphi}$
4. (e, n) JE PUBLIC KEY, (d, n) JE PRIVATE KEY

KEB MÁME SPRÁVU m , POTOM

$$P(m) = m^e \pmod{n}, \quad S(m) = m^d \pmod{n}$$

$$\text{ZREJME } P \circ S(m) = S \circ P(m) = m^{ed} \equiv m \pmod{n}.$$

ROZŠÍRENÍ EUKLIDOV ALGORITHMUS

EUKLID(a, b)

if $b = 0$ then return $(1, 0, a)$
else $(x', y', d) = \text{EUKLID}(b, a \bmod b)$
return $(y', x' - y' \lfloor \frac{a}{b} \rfloor, d)$

$$d = x'b + y'(a - \lfloor \frac{a}{b} \rfloor b) = y'a + (x' - y' \lfloor \frac{a}{b} \rfloor)b$$

POWER(a, b, n)

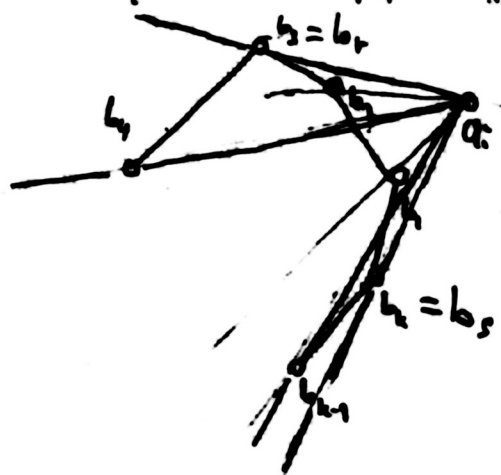
if $b = 0$ then return 1
else if $2 \mid b$ then return POWER($a \times a, \frac{b}{2}, n$)
else return $a \times \text{POWER}(a, b-1, n) \bmod n$

TINOG1 ALG. A DAT. STR. II - SKÚSKAKONVEKNÝ OBAL

PREDPOKLADAJEME, ŽE MÁME BODY $a_1, \dots, a_n$
ZOTRIEDENÉ PODĽA X-OVEJ SÚRADNICE

OZNAČME A_i KONVEKNÝ OBAL POLOST. Z BODOV $a_1, \dots, a_i$.
 $A_3 = a_3 a_2 a_1$ ALEBO $a_3 a_1 a_2$ (TAK AŽY BOL " ")

~ PREDP., ŽE MÁME KONV. OBAL $A_{i-1} = b_1 \dots b_k$
 b_1 JE $= a_{i-1}$ + JE ")



UVEDÚME POLPRÁNKY
 $a_i b_1, \dots, a_i b_k$

STERNICA NADSKÖR
KLEŠ' $\rightarrow$ RASTIE $\rightarrow$ KLEŠ'

NADČERNE b_i TAKÝ, ŽE $a_i b_i$ MÁ MAX. SMER.
 b_s TAKÝ, ŽE $a_i b_s$ MÁ MIN. SMER.

NOVÝ KONVEKNÝ OBAL $A_i = a_i b_1 b_{s+1} \dots b_{s-1} b_s$

BODU a_i ZAPOČÍTAN IBA NODRÉ PRÁNKY

TJ. $a_i b_1, a_i b_{s+1}, a_i b_s, a_i b_{s-1}$

PRÁNKY $a_i b_1 \dots a_i b_{s-1}, a_i b_s, a_i b_{s+1}$

ZAPOČÍTAVAN BODON $b_1, \dots, b_{s-1}, b_s, b_{s+1}, \dots, b_k$

$\Rightarrow$ KAŽDEMU BODU ZAPOČÍTAN NÁVNE 5 PRÁNKY

$\Rightarrow$ ČASOVÁ ZLOŽITOSŤ $O(n)$

LCS (LONGEST-COMMON-SUBSEQUENCE)

$a_1 \dots a_m$

$b_1 \dots b_n$

$$c_{ij} = \begin{cases} 1 + c_{i-1, j-1} & a_i = b_j \\ \max(c_{i-1, j}, c_{i, j-1}) & a_i \neq b_j \end{cases}$$

SORTING NETW.



$$d(S_1) = 0, d(M_1) = 1$$

$$c(S_1) = 0, c(M_1) = 1$$

$$d(S_n) = d(S_{n/2}) + d(M_{n/2})$$

$$c(S_n) = 2c(S_{n/2}) + c(M_{n/2})$$

$$d(M_n) = 1 + d(M_{n/2})$$

$$c(M_n) = n-1 + 2c(M_{n/2})$$

$$d(M_{2^k}) = k+1$$

$$d(S_{2^k}) = d(S_{2^{k-1}}) + k$$

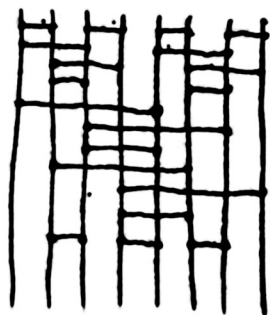
$$\Rightarrow d(S_{2^k}) = \frac{1}{2}k(k+1)$$

$$c(M_{2^k}) = 2^k - 1 + 2c(M_{2^{k-1}}) = (k+1)2^k - 1 - 2 - \dots - 2^{k-1} =$$

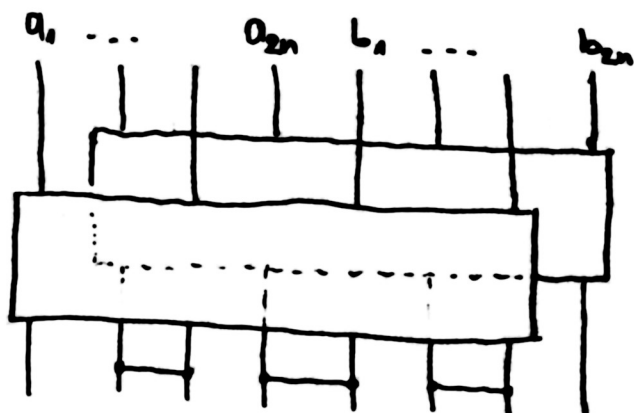
$$= (k+1)2^k - 2^k + 1 = k2^k + 1$$

$$c(S_{2^k}) = 2c(S_{2^{k-1}}) + (k-1)2^{k-1} + 1 =$$

$$= \frac{1}{2}k(k-1)2^{k-1} + 1 + \dots + 2^{k-1} = \frac{1}{2}k(k-1)2^{k-1} + 2^{k-1} - 1$$



TINOG1 ALG. A DAT. STR. II - SKÚŠKA



MAJME $a_1 < a_2 < \dots < a_{2n}$ $b_1 < b_2 < \dots < b_{2n}$
CHCEME SPRAVIT MERGE TÝCHTO ÚSEKOV

NAPR. PRE $n=2$ MAJME

$$a_1 < a_2 < a_3 < a_4 \quad b_1 < b_2 < b_3 < b_4$$

PO ZLATÍ MOŽE TO VZERAŤ NAPR. TAKTO

$$b_1 < b_2 < a_1 < b_3 < a_2 < a_3 < b_4 < a_4 \quad (1)$$

(STRIEDAVO VIBERAJEME ZO ZAČATKOV TÝCHTO POST.)

UŠŤOJNIME SI TERAZ DOLNÉ INDEXY PRVKOV

PRVÝ INDEX JE VŽDY $=1$ (BUĎ a_1 ALEBO b_1)

POSLEDNÝ INDEX JE VŽDY $2n$ (BUĎ a_{2n} ALEBO b_{2n})

NAVŤIŠE ŽIADNE TRI PO SEBE IDÚCE INDEXY NIE SÚ PÁRNE
A ŽIADNE 3 PO SEBE IDÚCE INDEXY NIE SÚ NEPÁRNE

OZNAČME O_i i -TY NASH. PRVOK Z PN. $\{a_1, a_3, \dots, a_{2n-1}, b_1, \dots, b_{2n-1}\}$
 E_i $-11-$ Z PN. $\{a_2, a_4, \dots, a_{2n}, b_2, \dots, b_{2n}\}$

POKROK NAŠA POST. (1) JE

$$O_1 \quad E_1 \quad O_2 \quad O_3 \quad E_2 \quad O_4 \quad E_3 \quad E_4$$

NIKDE NENAŠŤOBE 3 PO SEBE IDÚCE O ČKA / E ČKA
PRVÝ JE VŽDY O_1 A POSL. VŽDY E_{2n}

INDEXY PRI O -ČKACH A E -ČKACH MONOTÓNNE RASTÚ

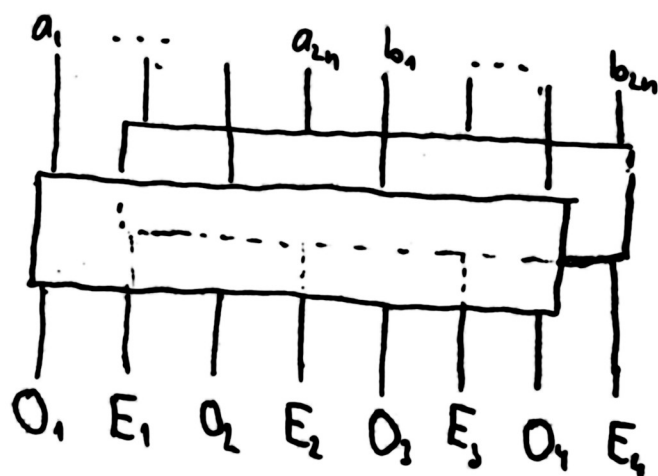
KĚJ NEUVĚDĚNĚ INDEXY, PŘIČEMŽ PŘÍK
 DE O_1 PO NĚJ NĚJ ÍST RUS
 DVOJICA $O E$ ALEBO $E O$
 PO TESTO DVOJICI NĚJ ÍST ŽASE IBA DVOJICA $O E$ V $E O$
 AŽ.

AŽ NA KONIEK PŘÍDE E

1. 2. 3.
 $O (E O) (O E) (O E) E$

→ V K-TEJ DVOJICI SA URČÍTE NACHÁZÍ
 E_k A O_{k+1}

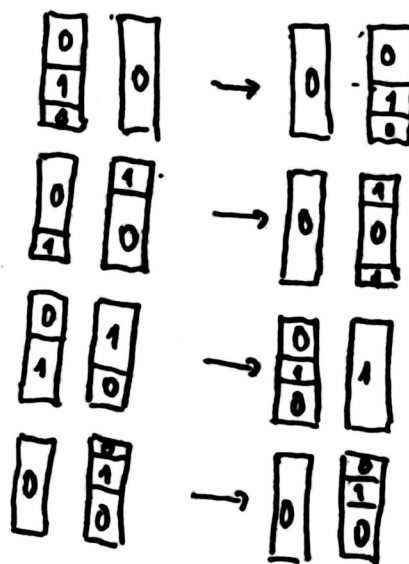
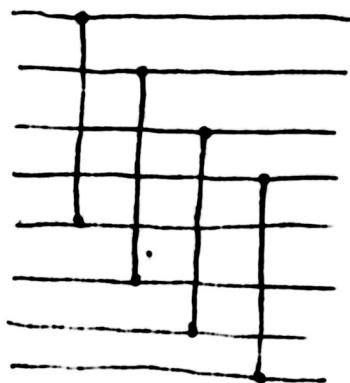
VÝSTUP 2 NERČERU



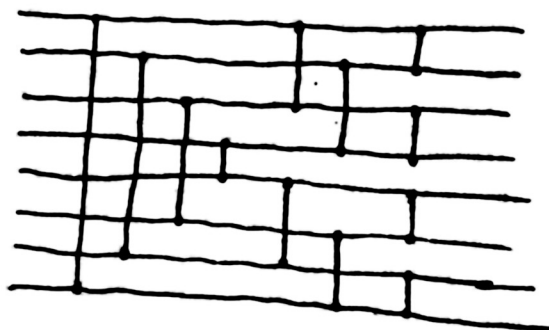
NA TO, AŽ JNE DOŠALI ZOTNĚDĚNÍ POSP.
 VE STACÍ IBA POROVNAT VNĚŠNÍ DVOJICE,
 D. $(E_1 O_2) (E_2 O_3) \dots (E_{2n-1} O_{2n})$

TINOG1 ALG. A DAT. STR. II - SKÚŠKA

BITONICKÉ TRIEDENIE



MERGE



SORT

