

PAL 1 AMORTIZOVANÁ SLOŽITOST. PRIORITNÍ FRONTY, HALDY (BINÁRNÍ, D-REGULÁRNÍ, BINOMIÁLNÍ, FIBONACCIHO), OPERACE NAD NIMI A JEJICH SLOŽITOST.

AGGREGATE ANALYSIS

- ZJISTIT NEJHORŠÍ ČAS PRO n OPERACÍ $T(n)$
- SLOŽITOST JE PAK $O(T(n))$
- PRŮMĚRNÁ CENA NA OPERACI, TEDY AMORTIZOVANÁ CENA NA OPERACI JE $\frac{O(T(n))}{n}$

STACK = LIFO

QUEUE = FIFO

MAX-HEAP PROPERTY

$H[PARENT(i)] \geq H[i]$

D-REGULÁRNÍ

TOLE INDEXOVANÉ OD 0 $\Rightarrow$

PŘEDEK k NA INDEXU $(k-1)/D$

POTOMCI $D \cdot k + 1, \dots, D \cdot k + D$

D MOCNINA 2 $\Rightarrow$ BINÁRNÍ POSUV

ACCOUNTING METHOD

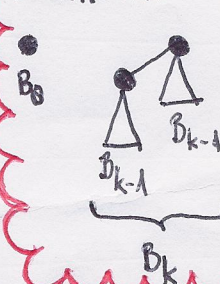
- CENA NA OPERACI (ACTUAL COST) = c_i
- AMORTIZOVANÁ CENA NA OPERACI = $\hat{c}_i$
- IF $\hat{c}_i - c_i > 0 \Rightarrow$ KREDIT
- KREDIT SE POUŽÍVÁ PŘEDĚL NA ZAPLACENÍ CENY ZA JINOU OPERACI
- VŽDY MUSÍ PLATIT KREDIT > 0 , TAK AMORTIZED COST, JE UPPER BOUND ACTUAL COST, TÍM P AMORTIZ. SLOŽITOST

POTENTIAL METHOD

- PODOBNĚ ACCOUNTING METHOD, ALE ŘEŠÍ SE POTENCIÁL CELÉ DATA STRUCTURE
- $\hat{c}_i = c_i + \phi(D_i) - \phi(D_{i-1})$

	BINARY (WORST)	BINOMIAL (WORST)	FIBONACCI (AMORTIZED)
MAKE-HEAP	$O(1)$	$O(1)$	$O(1)$
INSERT	$O(\lg n)$	$O(\lg n)$	$O(1)$
MINIMUM	$O(1)$	$O(\lg n)$	$O(1)$
EXTRACT-MIN	$O(\lg n)$	$O(\lg n)$	$O(\lg n)$
UNION	$O(n)$	$O(\lg n)$	$O(1)$
DECREASE-KEY	$O(\lg n)$	$O(\lg n)$	$O(1)$
DELETE	$O(\lg n)$	$O(\lg n)$	$O(\lg n)$

BINOMIAL TREE



BINOMIAL HEAP

- SET OF BINOMIAL TREES
- EACH TREE IN H IS MIN-HEAP-ORDERED, T.J. KEY OF NODE $\geq$ KEY OF ITS PARENT
- FOR INT $k \geq 0$, AT MOST 1 BINOMIAL TREE WITH ROOT'S DEGREE k IS IN H

ASYMPTOTICKÁ SLOŽITOST DEKLAROVÁNA NA ZÁKLADĚ NEJHORŠÍ (NEJLEPŠÍ) NOŽNÉ INSTANČE BĚHU PROGRAMU ALGORITMU, NICHOMĚ I NEJHORŠÍ SEKVENCE PŘÍPADŮ NŮŽE MÍT VÝRAZNĚ LEPŠÍ PRŮBĚH, PROTOŽE OPERACE S VYSOKOU SLOŽITOSTÍ ZMĚNÍ DATOVOU STRUKTURU TAK ŽE TAKTO ŠPATNÝ PŘÍPAD VEVSTÁVÁ PO KĚJAKOU DĚLSÍ DOBU A TÍM SE SLOŽITÁ OPERACE AMORTIZUJE

UNORDERED BINOMIAL TREE

- ROOT MÁ STUPĚŇ k (COŽ JE NEVYŠŠÍ STUPĚŇ)
- POTOMCI JSOU ROOTS PODSTROMKŮ V KĚJAKÉM PORADÍ (NA RORDIL OD BINOMIAL, KDE JSOU SEŘAZENÉ SESTUPNĚ PODLE STUPNĚ)

FIBONACCI HEAP

- SET OF MIN-HEAP-ORDERED TREES
- USPORÁDÁ DO BINOMIAL
- EXTRACT-MIN
 - ODEBERE MIN. Z ROOT LISTU
 - ~~2~~ 2 FIBONACCI HEAP VYTVOŘÍ BINOMIAL HEAP
- DECREASE-KEY, DELETE
 - ODEBERE NODE SVÁŽÍ MU KEY A PŘÍDA DO ROOT-LIST (KDYŽ KEY $\leq$ MIN, NAHRADÍ)
 - IF NODE DRUHÝ ODEBRANÝ SVÉHO PARENT $\Rightarrow$ PŘÍDA PARENT DO ROOT LIST, ATD...

PA2 EFEKTIVNÍ ALGORITMY PRO STANDARDNÍ GRAFOVÉ ÚLOHY: TARJANŮV A KOSARJU-SHARIRŮV ALGORITMUS
 DETEKCE SILNĚ SOUVISLÝCH KOMPONENT V ORIENTOVANÉM GRAFU. KONSTRUKCE TOPOLOGICKÉHO USPOŘÁDÁNÍ
 ACYKLICKÉHO GRAFU. ALGORITMY HLEDÁNÍ MINIMÁLNÍ KOSTRY GRAFU (BORŮKŮV, PRIMŮV, KRUSKALŮV). UNION-FIND PROBLÉM.

PROHLÉDÁVÁNÍ DO ŠÍŘY
 BFS - QUEUE
PROHLÉDÁVÁNÍ DO HLoubKY
 DFS - STACK
 1. VŠECHNY VRCHOLY UNVISITED
 2. DOKUD UNVISITED $\rightarrow v = \text{POP}()$
 3. $v.\text{STAV} = \text{OPEN}$
 4. FOREACH $n: v.\text{NEIGHBOURS}$
 PUSH(n)
 5. $v.\text{STAV} = \text{VISITED}$

TOPOLOGICKÉ USPOŘÁDÁNÍ PRO DAG
 V DFS, PŘI ZAVÍRÁNÍ NODE
 PO NÁVRATU Z REKURZE
 INKREMENTOVAT ČÍTAČ. POTOM
 PLATÍ, ŽE PREDECESSOR JE
 VŽDY VĚTŠÍ.

KOSARJU-SHARIRŮV ALGORITMUS
 IN: GRAF $G=(V,E)$
 OUT: SCC(G)
 1. $S = \text{EMPTY STACK}$
 2. WHILE $S \neq \text{CONTAIN ALL VERTICES}$
 $v \neq S$
 DFS(v), KDYŽ ZAVÍRÁ $v \rightarrow S.\text{PUSH}(v)$
 3. PŘEVRAŤIT SMĚR VŠECH HRAN
 (TRANSPONOVANÝ GRAF)
 4. WHILE $S \neq \text{EMPTY}$
 $v = S.\text{POP}()$
 IF $v == \text{UNVISITED}$ THEN DFS(v)
 NAŠTĚVENÉ VRCHOLY JSOU
 SILNĚ SOUVISLOU KOMPONENTOU
 OBSAHUJÍCÍ v .

TARJANŮV ALGORITMUS
 IN: GRAF $G=(V,E)$
 OUT: SCC(G)
 PRO VŠECHNY UNVISITED: find_scc(x)
 1. x/x
 2. $S.\text{PUSH}(x)$
 3. FOREACH $n: x.\text{SUCCESSOR}$
 IF UNVISITED, find_scc(n)
 $x/? = \min(x, n)$
 ELSE
 $x/? = \min(x, n)$
 FI
 4. IF $x/? == x$
 SCC++
 REPEAT
 ~~SCC~~ $y = S.\text{POP}()$
 SCC.PUSH(y)
 UNTIL $x == y$
 FI

MST - MINIMUM SPANNING TREE
PRIMŮV (JARNÍKŮV) ALGORITMUS
 1. $MST = \emptyset$
 2. VYBER LÍBOVOLNÝ VRCHOL
 3. VYBER NEJLEVNĚJŠÍ HRANU
 SPOJUJÍCÍ VYBRANÝ VRCHOL
 S VRCHOLEM, ~~KTERÝ JE~~ KTERÝ MENÍ V MST
 4. PŘIDEJ HRANU DO MST

BORŮKŮV ALGORITMUS
 - VŠECHNY CENY HRAN RŮZNÉ
 1. WHILE $|SCC(G)| \geq 2$
 VYBER NEJLEVNĚJŠÍ HRANU
 PROPOJUJÍCÍ VŽDY 2 KOMPONENTY
 2. TYTO HRANU PŘIDEJ DO MST

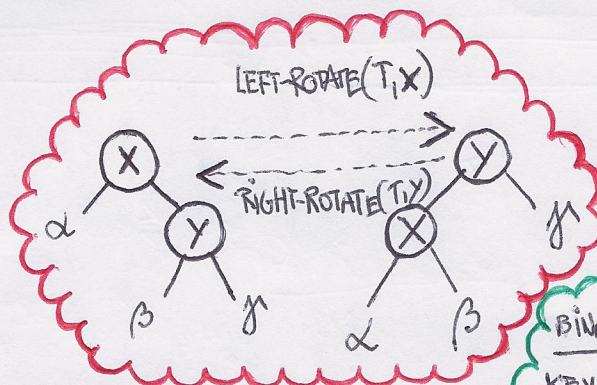
KRUSKALŮV (GREEDY) ALGORITMUS
 1. SEŘADIT HRANU PODLE CENY VZESTUPNĚ
 2. $MST = \emptyset$
 3. FOR $i=1 \dots |E(G)|$
 IF $MST \cup \{e_i\}$ IS ACYCLIC THEN
 $MST \cup = \{e_i\}$

UNION-FIND PROBLÉM
 Q: PATŘÍ u A v DO STEJNÉ PROPOJENÉ KOMPONENTY?
JEDNODUCHÉ ŘEŠENÍ
 - VŠECHNY VRCHOLY $1..n$
 - POLE $R[1..n]$ OBSAHUJE IDENTIFIKÁTOR SCC
 - $\text{FIND}(v) = R[v]$
 - $\text{UNION}(u,v) \rightarrow$ POKUD $R[u] \neq R[v]$, VŠECHNY $R[v]$ PŘEPAT NA $R[u]$
VYLEPŠENÉ (DIRECTED TREE) ŘEŠENÍ
 - KAŽDA KOMPONENTA ULožENA JAKO DIRECTED TREE
 - ROOT JE IDENTIFIKÁTOR SCC
 - $\text{FIND}(v) \rightarrow$ PŘES POINTERY NA PARENT ZJIŠTÍ ROOT
 - $\text{UNION}(u,v) \rightarrow$ POKUD $\text{ID}(u) \neq \text{ID}(v)$, MENŠÍ SCC VLOŽENA DO TREE VĚTŠÍ

AL3 VYHLEDÁVACÍ STROMY: B, B+, R-B, 2-3-4, SPLAY A JEJICH PRAKTICKÉ VYUŽITÍ, PROBLEMATIKA VYHLEDÁVÁNÍ VE VÍCE DIMENZÍCH, K-D STROMY.

R-B TREE (BALANCED BINARY SEARCH TREE)

- PROPERTIES:
1. EVERY NODE **R** OR **B**
 2. ROOT IS **B**
 3. LEAF (NIL) IS **B**
 4. IF NODE IS **R**, BOTH LEAVES **B**
 5. EACH NODE VŠECHNY ČESTY K POTOMKŮM STŘINÝ POČET **B** NODES
- WEIGHT AT MOST $2 \cdot \lg(n+1) = O(\lg n)$



BINARY SEARCH TREE PROPERTIES

KEY IN EACH NODE MUST BE GREATER THAN ALL KEYS STORED IN THE LEFT SUB-TREE AND SMALLER THAN ALL KEYS IN RIGHT SUB-TREE

B-TREE

- POUŽITÍ V HDD
- ZABECNENÍ BIN. SEARCH TREE
- KEY AND VALUE STORED IN NODE/LEAF
- ALL LEAVES SAME DEPTH
- EVERY INTERNAL NODE AT LEAST t CHILDREN
- EVERY INTERNAL NODE AT MOST $2t$ CHILDREN

B+ TREE

- B-TREE
- VALUES POUZE V LEAVES (VĚTŠINOU PŘÍDAVÁ 1 VRSOVA S HODNOTAMI)

2-3-4 TREE

- B-TREE stupně 4 (PZN. $t=2$)
- NA IMPLEMENTACI SLOŽNĚJŠÍ NEŽ R-B TREE (VÍCE SPECIAL CASES)

SPLAY TREE

- BINARY SEARCH TREE
- KADA OPERACE KOMBINOVANÁ S OPERACÍ SPLAY
- ZAJISTI, ŽE NODE SE STANE ROOTEM STROMU (POMOCÍ ROTATIONS)
- ČASTO DOTAZOVANÉ NODES JSOU ULEVPE DOSTUPNÉ
- READ-ONLY ACCESS TAKÉ MĚNÍ STRUKTURU $\Rightarrow$ PROBLÉM PŘI MULTITHREADINGU
- 3 TYPES OF STEPS:
 - zig: IF $p \neq \text{ROOT}$, ROTATION TO MAKE x ROOT
 - zig-zig: IF $p \neq \text{ROOT}$ & x, p both left-child or right-child, 2 ROTATIONS (SAME DIRECTION) TO MAKE x ROOT
 - zig-zag: IF $p \neq \text{ROOT}$ & x, p diff. -childs, 2 ROTATIONS (DIFF. DIRECTION) TO MAKE x ROOT
- USEFUL FOR IMPLEMENTING CACHED, GARAGE COLLECTIONS

VYHLEDÁVÁNÍ VE VÍCE DIMENZÍCH

- PRO VYHLEDÁNÍ INTERFACE V DIMENZÍ DIM MUSÍM VYHLEDÁVAT VE VŠECH VĚTVÍCH VŠECH OSTATNÍCH DIMENZÍ, ROZHODNUTÍ (VĚTVENÍ) MŮŽE PROČEST POUZE V DIMENZÍ, VE KTERÉ VYHLEDÁVÁM

K-D TREE

- DELETE POUZE LEAVES, VNITŘNÍ NODES SE NAHRADÍ VHDNÝM
- K-DIMENZIONÁLNÍ TREE, DIMENZE SE STRIDAJÍ PO ÚROVNÍCH
- NEAREST NEIGHBOUR SEARCH: HLEDÁ NEJBLIŽŠÍ NODE K ZADANÉMU
- RANGE SEARCH: HLEDÁ ROZSAH PARAMETRŮ
- BUILD STATIC K-D TREE: $O(n \cdot \lg n)$
- INSERT A NEW POINT: $O(\lg n)$
- REMOVE POINT: $O(\lg n)$
- RANGE: $O(n^{1/k} + m)$
- FIND 1 NEAREST NEIGHBOUR: $O(\lg n)$

$$A = \{q \in Q \mid q \cap A_2 \neq \emptyset\}$$

TAL5 ALGORITHMUS, SPRÁVNOST ALGORITHMU, SLOŽITOST ALGORITHMU, SLOŽITOST ÚLOHY, TŘÍDA P, TŘÍDA NP

DOBŘE DEFINOVANÝ PROCES (POSLUPOVNOST VÝPRAVNÝCH KROKŮ), KTERÝ PŘIJÍMÁ HODNOTY (VSTUP) A VYTVAŘÍ HODNOTY (VÝSTUP)

TŘEBA Ověřit (SPRÁVNOST)

1. ALGORITHMUS SE NA KAŽDÉM VSTUPU ZASTAVÍ **VARIANT**
2. ALGORITHMUS PO ZASTAVENÍ VYDÁ SPRÁVNÝ VÝSTUP **INVARIANT**

$$O(g(n)) = \{f(n) \mid \exists c > 0, n_0 \text{ tak, že } f(n) \leq c \cdot g(n) \forall n \geq n_0\}$$

$$\Omega(g(n)) = \{f(n) \mid \exists c > 0, n_0 \text{ tak, že } f(n) \geq c \cdot g(n) \forall n \geq n_0\}$$

$$\Theta(g(n)) = \{f(n) \mid \exists c_1, c_2 > 0, n_0 \text{ tak, že } c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n) \forall n \geq n_0\}$$

$$o(g(n)) = \{f(n) \mid \forall c > 0 \exists n_0 \text{ tak, že } 0 \leq f(n) < c \cdot g(n) \forall n > n_0\}$$

$$\omega(g(n)) = \{f(n) \mid \forall c > 0 \exists n_0 \text{ tak, že } 0 \leq c \cdot g(n) < f(n) \forall n > n_0\}$$

MASTER THEOREM (ŘEŠENÍ REKURSVNÍCH VZTAHŮ)

$$T(n) = aT\left(\frac{n}{b}\right) + f(n) \quad a \geq 1, b > 1$$

1. IF $f(n) \in O(n^{\log_b a - \epsilon})$ pro nějak. konst. $\epsilon > 0$, pak $T(n) \in \Theta(n^{\log_b a})$
2. IF $f(n) \in \Theta(n^{\log_b a})$, pak $T(n) \in \Theta(n^{\log_b a} \lg n)$
3. IF $f(n) \in \Omega(n^{\log_b a + \epsilon})$ pro nějak. konst. $\epsilon > 0$
 a IF $a \cdot f\left(\frac{n}{b}\right) \leq c \cdot f(n)$ pro nějak. konst. $c < 1$
 pro všechna dost. velká n , pak $T(n) \in \Theta(f(n))$

IF $f(n) \in \Theta(n^{\log_b a} \lg^k n)$ pro $k \geq 0$, pak $T(n) \in \Theta(n^{\log_b a} \lg^{k+1} n)$

ÚLOHY ROZHODOVACÍ, VYHODNOCOVACÍ, OPTIMALIZAČNÍ

SLOŽITOST ALGORITHMU - POČET KROKŮ TM
 SLOŽITOST ÚLOHY - ČASOVÁ, PROSTOROVÁ

TŘÍDA P: ROZHODOVACÍ ÚLOHA U LEŽÍ VE TŘÍDĚ P , JESTLIŽE EXISTUJE DETERMINISTICKÝ TURINGŮV STROJ, KTERÝ ROZHODNE JAZYK $L_U \rightarrow$ SKLÁDÁ SE ZE VSECH SLOV ODPOVÍDAJÍCÍCH ANO-INSTANCÍM V POLYNOMIÁLNÍM ČASE, T.J. $T(n) \in O(p(n))$ PRO NĚJAKÝ POLYNOM $p(n)$

TŘÍDA NP: ROZHODOVACÍ ÚLOHA U LEŽÍ VE TŘÍDĚ NP , JESTLIŽE EXISTUJE NEDETERMINISTICKÝ TURINGŮV STROJ, KTERÝ ROZHODNE JAZYK L_U V POLYNOMIÁLNÍM ČASE

NEDETERMINISTICKÝ ALGORITHMUS

1. NÁHODNĚ VYGENERUJE ŘETĚZEC s
2. DETERMINISTICKÝ ALGORITHMUS NA ZÁKLADĚ VSTUPU A A s VRÁTÍ ODPOVĚď ANO NEBO NEVÍM

SLOŽITOST BODŮ $O(P(n))$

NEDETERMINISTICKÝ ALGORITHMUS ŘEŠÍ U , IF

1. PRO KAŽDOU ANO INSTANCI u EXISTUJE s , NA JEHOŽ ZÁKLADĚ DÁ ALGORITHMUS ANO
2. PRO ŽÁDNOU NE INSTANCI u NEEXISTUJE s , NA JEHOŽ ZÁKLADĚ DÁ ALGORITHMUS ANO

VSTUPNÍ SLOVO $w \in \Sigma^*$ JE PŘIJATO TURINGOVÝM STROJEM PRAVĚ TEHDY, KDYŽ SE TM NA SLOVĚ w ÚSPĚŠNĚ ZASTAVÍ

YNŮŽINA SLOV $w \in \Sigma^*$ KTERÉ TM PŘIJÍMÁ JE JAZYK PŘIJÍMANÝ TM, ZNACÍ SE $L(M)$

TM ROZHODUJE JAZYK L KDYŽ HO PŘIJÍMÁ A NA KAŽDÉM VSTUPU SE ZASTAVÍ

REDUKCE, POLYNOMIÁLNÍ REDUKCE

DÁNY ROZHODOVACÍ ÚLOHY U A V
 U SE REDUKUJE NA V , KDYŽ

EX. ALG. (TM), KTERÝ PRO KAŽDOU
 INSTANCI U ÚLOHY U ZKONSTRUJE
 INSTANCI V ÚLOHY V TAK, ŽE

U JE AND-INSTANCI U IFF V JE AND-INSTANCI V

ZNAČÍME $U \leq V$

POLYNOMIÁLNÍ REDUKCE

JESTLI ŽE ALG. PRACUJE V POLYNOM.

CASE PAK $U \leq_p V$

NPC (NP-ÚPLNÉ)

ŘEKEME ŽE ROZHODOVACÍ ÚLOHA U JE
 NP ÚPLNÁ, JESTLI ŽE

1. U JE VE TŘÍDĚ NP
2. KAŽDÁ NP ÚLOHA SE POLYNOMIÁLNĚ
 REDUKUJE NA U

NP-TĚŽKÉ

D ÚLOZE U VÍME JEN, ŽE SE NA NÍ
 POLYNOMIÁLNĚ REDUKUJE NĚKTERÁ
 ÚLOHA Z NPC, PAK U JE NP-TĚŽKÁ

ŘEŠENÍ NPC

MALE INSTANČE $\Rightarrow O(e^n)$ OK

IZOLOVAT SPECIAL CASES $\rightarrow O(p(n))$

NEAR OPT. SOLUTION $\rightarrow$ APROXIMAČNÍ ALG.

RTM - RANDOMIZED TM

JF TM, KTERÝ MÁ NAVÍC RO PÁSKU
 S NÁHODNĚ GEN. 0 A 1, KDE $pr(0) = pr(1) = \frac{1}{2}$

TŘÍDA RP

JAZYK L PATŘÍ DO RP IFF EX. RTM M , ŽE:

1. IF $w \in L$, M SE V q_f ZASTAVÍ S PR. 1
2. IF $w \notin L$, M SE V q_f ZASTAVÍ S PR. ALESPŮ $\frac{1}{2}$
3. EX $p(n)$ ŽE BĚH M (PRO LIB. OBSAH DRUHÉ
 PÁSKY) TRVÁ MAX. $p(n)$ KROKŮ

MALE FERMATOVA VĚTA

A KESOUČNĚ $3, p$

$$a^{p-1} \equiv 1 \pmod{p}$$

COOKEOVA VĚTA

ÚLOHA SAT (SPLNITELNOST FORMULE V KNF-
 KONJUNKTIVNÍ NORMÁLNÍ FORMA) JE NP-ÚPLNÁ

NEDETERMINISTICKÝ ALGORITMUS (NTM)

1. VYGENERUJE OHODNOCENÍ LOG. PROM.
2. POPIS PRÁCE TM FORMULÍ VÝROK. LOG.

PŘÍKLAD REDUKCE: SAT $\leq_p$ 3-CNF SAT
 3-BAREVNOST $\leq_p$ ILP

HEURISTIKY

PRO VĚTŠÍ INSTANČE NPC ÚLOH $\Rightarrow$ POUZE "DOSTATEČNĚ PŘESNÉ" ŘEŠENÍ

HEURISTICKÉ ALGORITMY V POLYNOMIÁLNÍM ČASE

APROXIMAČNÍ ALGORITMY - UMÍME UŘČIT, JAK DALEKO OD OPTIMÁL.

TSP HEURISTIKA

— SPLŮJE Δ NEROVNOST

INSTANCI I — ÚPLNÝ GRAF G S MNOŽ. VRCHOLŮ $V = \{1, 2, \dots, n\}$
 A OHODNOCENÍM d

1. V G NAJÍT MÍNIMÁLNÍ KOSTRU (V, K)
2. (V, K) PROHLÉDAEME DO HLUBKY 2 LIB. VRCHOLU
3. TRASA T TAK, ŽE VRCHOLY PROJÍT VE STEJNÉM
 POŘADÍ JAKO PŘI PRVNÍM NAVŠTÍVENÍ BĚHEM PROHLÉDAVÁNÍ

DĚLKA TRASY $\leq 2 \cdot \text{OPT}(I)$

TSP HEURISTIKA - CHRISTOFIDESŮV ALGORITMUS

1. V G NAJÍT MÍNIMÁLNÍ KOSTRU (V, K)
2. ÚPLNÝ GRAF H NA MNOŽINĚ VŠECH VRCHOLŮ,
 KTERÉ V (V, K) MAJÍ LICHÝ STUPEŇ
3. V H NEJLEVNĚJŠÍ PERFEKTNÍ PÁROVÁNÍ P
4. HRANY P PŘIDAT K HRANÁM K . GRAF $(V, P \cup K)$
 JE EULEROVSKÝ, SESTROJÍME UZAVŘENÝ EULEROVSKÝ
 TAH
5. TRASA T TAK, ŽE VRCHOLY V POŘADÍ, VE KTERÉM
 POPRVÉ VSTOUPÍ PŘI TVORBĚ EULER. TAHU

DĚLKA TRASY $\leq \frac{3}{2} \cdot \text{OPT}(I)$

MILLERŮV TEST PRVOČISELNOSTI

VSTUP: VELKÉ LICHÉ PŘÍRODNÍ ČÍSLO n

1. SPČÍTÁME $n-1 = 2^l \cdot m$, m — LICHÉ
2. NÁHODNĚ $a \in \{1, 2, \dots, n-1\}$
3. SPČÍTÁME $a^m \pmod{n}$, IF $a^m \equiv 1 \pmod{n} \rightarrow$ "PRVOČÍSLO"
4. OPAK. OHODNŮV. $a^{2^l \cdot m} \pmod{n}$, $a^{2^{l-1} \cdot m} \pmod{n}, \dots, a^{2^1 \cdot m} \pmod{n}$
5. IF $a^{2^k \cdot m} \not\equiv 1 \pmod{n} \rightarrow$ "SLOŽENÉ"
6. K TAKOVĚ, ŽE $a^{2^k \cdot m} \not\equiv 1 \pmod{n}$ & $a^{2^{k+1} \cdot m} \equiv 1 \pmod{n}$
 IF $a^{2^k \cdot m} \equiv -1 \pmod{n} \rightarrow$ "PRVOČÍSLO"
 IF $a^{2^k \cdot m} \not\equiv -1 \pmod{n} \rightarrow$ "SLOŽENÉ"

IF MILLER(n) == "SLOŽENÉ"
~~IF~~ n SLOŽENÉ

IF MILLER(n) == "PRVOČÍSLO"
 $pr(n \text{ PRVOČÍSLO}) > \frac{1}{2}$

TAT TURINGOVY STROJE, REKURZIVNÍ A REKURZIVNĚ SPÖETNĚ JAZYKY, ALGORITMICKY NĚŘŠITELNĚ ŮLOHY

- ŘÍDÍCÍ JEDNOTKA (V JEDNOM Z KONEČNĚ MNOHA STAVŮ)
- POTENCIÁLNĚ NEKONEČNÁ PÁSKA (ROZDĚLENÁ NA POLE)
- HLAVA UMOŽŇUJÍCÍ ČÍST A/NEBO PŘEPÍŠOVAT POLE

RAM - POČÍTAČ S LÍBOVOLNÝM PŘÍSTUPEM
JEDNOTKY: PROGRAMOVÁ, ARITHMETICKÁ, PAMĚT, VSTUPNÍ, VÝSTUPNÍ

FORMÁLE: SEDMICE $(Q, \Sigma, \Gamma, \delta, q_0, B, F)$

- Q - KONEČNÁ MNOŽINA STAVŮ
- Σ - KONEČNÁ MNOŽINA VSTUPNÍCH SYMBOLŮ
- Γ - KONEČNÁ MNOŽINA PÁSKOVÝCH SYMBOLŮ, $\Sigma \subset \Gamma$
- B - PRAZDINÝ SYMBOL, $B \in \Gamma \setminus \Sigma$

δ - PŘECHODOVÁ FCE, PARC. ZOB. $(Q \setminus F) \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$

$q_0 \in Q$ JE POČÁTEČNÍ STAV

$F \subseteq Q$ JE MNOŽINA KONEČNÝCH STAVŮ

SITUACE TM: $x_1 x_2 \dots x_{i-1} q_i x_{i+1} \dots x_k$

KROK TM: $x_1 x_2 \dots x_{i-1} y_p x_{i+1} \dots x_k$
ČÁLEŽI NA δ , IF NEDEFINOVÁNO, TM SE ZASTAVÍ

VÝPOČET TM: NAD SLOVEM w , POLOUPNOST KROKŮ,
IF TM ZASTAVÍ V KONEČNÉM STAVU,
ZASTAVÍ SE ÚSPĚŠNĚ, VÝSTUP NA PÁSCĚ

$w \in \Sigma^*$ JE PŘIJATO TM, KDYŽ SE TM NA w ÚSPĚŠNĚ ZASTAVÍ
MNOŽINU SLOV $w \in \Sigma^*$ KTERÉ TM PŘIJÍMÁ JE JAZYK PŘIJÍMANÝ
TM, ZNAČÍME $L(TM)$

TM ROZHODUJE L , JESTLIŽE TENTO JAZYK PŘIJÍMÁ
A NAVÍC SE NA KAŽDÉM VSTUPU ZASTAVÍ

NEDETERMINISTICKÝ TURINGŮV STROJ

$\delta: (Q \setminus F) \times \Gamma \rightarrow P_f(Q \times \Gamma \times \{L, R\})$

$P_f(X)$ JE KONEČNÁ RODMNOŽINA MNOŽINY X

TRÍDA RP (PRO RTM)

1. IF $w \notin L$, RTM SE ZASTAVÍ V q_f S TRAVĚPOD. = 0
2. IF $w \in L$, RTM $\rightarrow q_f$ $\rightarrow$ $\geq \frac{1}{2}$
3. $\exists p(n)$, ŽE KAŽDÝ BĚH RTM TRVÁ MAX. $p(n)$ KROKŮ PRO VSTUP DĚLKY n

MONTE CARLO TM: SPLŇUJE PODMÍNKY 1,2

TRÍDA ZPD (PRO RTM)

1. IF $w \notin L$, ZASTAVÍ V q_f S PŠTÍ 0
2. IF $w \in L$, $\rightarrow$ $\rightarrow$ 1
3. STŘEDNÍ HODNOTA POČTU KROKŮ JE $p(n)$

LAS-VEGAS TM: SPLŇUJE 1,2,3

REKURZIVNÍ JAZYKY: IF $\exists TM$, KTERÝ ROZHODUJE L , PAK L REKURZIVNÍ

REKURZIVNĚ SPÖETNĚ JAZYKY: IF $\exists TM$, KTERÝ PŘIJÍMÁ L , TAK L REKURZIVNĚ SPÖETNÝ
JAZYKY, KTERÉ NEJSOU REKURZIVNÍ, ŘÍKÁME ALGORITMICKY NĚŘŠITELNĚ

KÓD TM

DIAGONÁLNÍ JAZYK L_d

UNIVERZÁLNÍ JAZYK L_u : MNOŽINA SLOV $\langle M \rangle \# w$; $\langle M \rangle \sim$ KÓD TM, $w \in \{0,1\}^*$, $w \in L(M)$

UNIVERZÁLNÍ TM: PŘIJÍMÁ L_u , ZNAČÍME U

~~4~~ - 4 PÁSKY

NEROZHODNUTELNĚ ŮLOHY - PŘ:

PCP - POSTOV KORESPONDENČNÍ PROBLÉM

$L_u \triangleleft MPCP \triangleleft PCP$

KOB METODA VĚTVÍ A MEZI. ALGORITMY PRO ILP. FORMULACE OPTIMALIZAČNÍCH A ROZHODOVACÍCH PROBLÉMŮ

POMOCÍ ILP. TOKY A ŘEZY. MULTI-KOMODITNÍ TOKY.

VÝČETNÁ METODA (ENUMERATIVE)

- PROZKOUMÁNÍ VSECH MOŽNÝCH ŘEŠENÍ
- VYBRAT OPTIMÁLNÍ

METODA VĚTVÍ A MEZI

- VYŘEŠÍ LP PROBLÉM. IF $x_i \in \mathbb{Z}$ PROVI $\rightarrow$ RETURN
- VYBER NĚJAKÉ $x_i \notin \mathbb{Z}$, HODNOTU PŘÍDEJ DO k
- ROZDĚLIT SOLUTION SPACE NA $x_i \leq \lfloor k \rfloor$ A $x_i \geq \lfloor k \rfloor + 1$
- REKURZIVNĚ OPRAKOVAT, DOKUD NENALEZENO VHDNĚ INT. ŘEŠENÍ ($x_i \in \mathbb{Z}$ PRO $\forall i$)
- UCHOVÁVÁ SE AKTUÁLNÍ NEJLEPŠÍ ŘEŠENÍ
- POKUD ŘEŠENÍ VYPOČTENÉ V AKTUÁLNÍ VĚTVI JE HORSÍ, VĚZ PAK KONČIT (MEZNÍ VĚTV)

CUTTING-PLANE METHOD

- ŘEŠENÍ POMOCÍ OPRAKOVÁNÍ LP
- PŘIDÁVÁNÍ PODMÍNEK OMEZUJÍCÍCH SOLUTION SPACE, ALE VŠECHNA INT. ŘEŠENÍ Z MINULÉHO KROKU MUSÍ ZŮSTAT
- PODMÍNKY SE NAZÝVAJÍ: DANTZIG CUTS
GOMORY CUTS
CHVÁTAL-GOMORY CUTS

ALGORITHMUS:

1. VYŘEŠIT PROBLÉM JAKO LP (SIMPLEX ALG.)
2. IF ŘEŠENÍ INT. $\rightarrow$ RETURN
3. PŘIDAT PODMÍNKU (GOMORY CUT) DO SIMPLEX TAB. OPTIMALIZOVAT BY DUAL LP, GOTO 2

FORD-FULKERSON ALGORITHMUS

- FORWARD ARC - HRANA ORIENTOVANÁ SPRÁVNÝM SMĚREM JAKO CESTA $s \rightarrow t$
- BACKWARD ARC - HRANA ORIENTOVANÁ OPACNĚ
- AUGMENTING PATH - NEORIENTOVANÁ CESTA $s \rightarrow t$, KDE $f(e) < u(e)$ PRO FORWARD, $f(e) > l(e)$ PRO BACKWARD
- KAPACITA AUGM. PATH - NEJVYŠŠÍ MOŽNÉ ZVÝŠENÍ TOKU

1. NADJI FEASIBLE FLOW $f(e)$ PRO VŠECHNY $e \in E(G)$
2. NADJI AUGMENTING PATH P , KDYŽ NENÍ STOP.
3. VYPOČET KAPACITU, ZVĚTŠIT TOK, GOTO 2.

FINDING AUGMENTING PATH

1. $m_v = \text{FALSE}$ $\forall v \in V(G)$, $m_s = \text{TRUE}$
2. IF $\exists e \in E(G)$, $e = (v_i, v_j)$, $m_i = \text{TRUE}$ & $m_j = \text{FALSE}$ & $f(e) < u(e) \Rightarrow m_j = \text{TRUE}$
3. IF $\exists e \in E(G)$, $e = (v_i, v_j)$, $m_i = \text{FALSE}$ & $m_j = \text{TRUE}$ & $f(e) > l(e) \Rightarrow m_i = \text{TRUE}$
4. IF t REACHED, AUGM. PATH FOUND, STOP
IF NOT POSSIBLE MARK ANOTHER NODE, AUGM. PATH NEEK, STOP $\rightarrow$ AKTUÁLNĚ OZNACENÉ VRCHOLY JSOU MINIMUM CUT
ELSE GOTO 2 NEBO 3

FINDING INITIAL FLOW WITH NONZERO LOWER BOUNDS CAN BE TRANSFORMED TO FEASIBLE FLOW DECISION AND THEN MAX. FLOW PROBLEM. $\rightarrow t \rightarrow s$ (w/ CAPACITY), SUBST. $f(e) = l(e) + l(e)$, s', t' , SOLVE MAX. FLOW

MATRIX $A \in \mathbb{R}^{m \times n}$

VECTORS $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$

GOAL IS TO FIND $x \in \mathbb{Z}^n$

SUCH THAT $A \cdot x \leq b$

AND $c^T \cdot x$ IS THE MAX

$\max \{c^T \cdot x : A \cdot x \leq b, x \in \mathbb{Z}^n\}$

PRO ROZHODOVACÍ PROBLÉMY MÍSTO MAXIMALIZACE $c^T \cdot x$ STAČÍ JEN 1, ŘEŠÍME, ZDA JE ILP SPLNITELNÉ

SÍŤ

JE PĚTICE (G, l, u, s, t) , KDE

G : ORIENTOVANÝ GRAF

$l: E(G) \rightarrow \mathbb{R}_0^+$ MINIMÁLNÍ KAPACITA

$u: E(G) \rightarrow \mathbb{R}_0^+$ MAXIMÁLNÍ KAPACITA

s : ZDROJOVÝ UZEL (SOURCE)

t : SPOTŘEBNÍ UZEL (SINK)

SÍŤOVÝ TOK

$f: E(G) \rightarrow \mathbb{R}_0^+$ JE TOK, POKUD PRO KAŽDÝ UZEL KROMĚ s A t PLATÍ KIRCHHOFFŮV ZÁKON:

$$\sum_{e \in \delta^-(v)} f(e) = \sum_{e \in \delta^+(v)} f(e)$$

FEASIBLE FLOW

$f(e) \in \langle l(e), u(e) \rangle$

PRO $l(e) > 0$ NEMUSÍ EXISTOVAT

MAXIMUM FLOW

CÍL JE PRO (G, l, u, s, t) NADJI FEASIBLE

FLOW, KTERÝ MAX. $\sum_{e \in \delta^+(s)} f(e) - \sum_{e \in \delta^-(s)} f(e)$

MAXIMUM FLOW AS LP

$f(e) \in \mathbb{R}_0^+$ PŘEDSTAVUJE TOK HRANOU $e \in E(G)$

$\max \sum_{e \in \delta^+(s)} f(e) - \sum_{e \in \delta^-(s)} f(e)$

s.t. $\sum_{e \in \delta^-(v)} f(e) = \sum_{e \in \delta^+(v)} f(e)$ $\forall v \in V(G) \setminus \{s, t\}$

$l(e) \leq f(e) \leq u(e)$ $e \in E(G)$

MINIMUM CUT

(MNOŽINA HRAN $\delta(A)$, $s \in A$; $t \in V(G) \setminus A$)

$\rightarrow$ MÁ MINIMÁLNÍ KAPACITU $c(A) = \sum_{e \in \delta^+(A)} u(e) - \sum_{e \in \delta^-(A)} l(e)$

MINIMUM COST FLOW AS LP

$\min. \sum_{e \in E(G)} c(e) \cdot f(e)$

s.t. $\sum_{e \in \delta^-(v)} f(e) - \sum_{e \in \delta^+(v)} f(e) = b(v)$ $\forall v \in V(G)$

$l(e) \leq f(e) \leq u(e)$

MINIMUM COST MULTICOMMODITY

$\min. \sum_{e \in E(G)} \sum_{m \in M} f^m(e) \cdot c(e)$

s.t. $\sum_{e \in \delta^-(v)} f^m(e) - \sum_{e \in \delta^+(v)} f^m(e) = b^m(v)$ $\forall v \in V(G) \forall m \in M$

$l(e) \leq \sum_{m \in M} f^m(e) \leq u(e)$

KO9 NEJKRATŠÍ CESTY. ÚLOHA ČESKÉHO CESTOVÁNÍ. HEURISTIKY A APROXIMAČNÍ ALGORITMY. METODA DYNAMICKÉHO PROGRAMOVÁNÍ. PROBLÉM BATOKU. PSEUDO-POLYNOMIÁLNÍ ALGORITMY.

SPT

- DIGRAPH G , COST
- CIL: NEJKRATŠÍ CESTA $s-t$
 - s - všechny vrcholy
 - všechny vrcholy $-t$
 - mezi všemi vrcholy
- NEGATIVE WEIGHT - OK
- NEGATIVE CIRCUIT - NO!
- ROZCHLENKOVANÁ NEROVNOST

BELLMANŮV PRINCIP OPTIMALITY
 P nejkratší $s-w$ cesta
 $e=(v,w)$ je poslední hranou
 pak $P[s,v]$ nejkratší $s-v$ cesta
 s max. $k-1$ hran

DIJKSTRA

- BFS, VYBER NODE S NEJMENŠÍ COST
- ~~POKUD SE~~ (LABEL)
- VYPOČET COST PRO CHILD NODES
 2 AKTUÁLNÍHO NODE
- POKUD MENŠÍ, NAHRAD
- KDYŽ COST PRO VŠECHNY CHILD, TAK

BELLMAN-FORD

- FOR $i=1..n-1$ (v,w)
- FOR EACH $e \in E(G)$
 - IF $len(w) > len(v) + C(v,w)$
 - FI $len(w) = len(v) + C(v,w)$

FLOYD (WARSHALL)

- FOR ALL k CHECK IF l_{ij} IMPROVES
- FOR $k=1..n$
 - FOR $i=1..n$
 - FOR $j=1..n$
 - IF $l_{ij} > l_{ik} + l_{kj}$
 - FI $l_{ij} = l_{ik} + l_{kj}$

DYNAMICKÉ PROGRAMOVÁNÍ

- OPTIMAL SUBSTRUCTURE = OPTIMÁLNÍ ŘEŠENÍ POMOCÍ KOMBINACÍ OPTIMÁLNÍCH ŘEŠENÍ PODPROBLÉMU
- OVERLAPPING SUBPROBLEMS: ŘEŠENÍ PODPROBLÉMU SE UKLÁDÁ A POUŽÍVÁ, KÝŽ JE POTŘEBA

KNAPSACK

- n : počet
- c_i : cena
- w_i : váha
- W : max. váha
- Z : max. cenu, $\leq W$
- COMPLEXITY REDUCTION BY ROUNDING DATA
- VŠECHNY CENY VYDĚLIT 2 A ZAKROUHLIT DOLŮ

SOLUTION FOR FRACTIONAL

- $\frac{c_i}{w_i}$: RELATIVE COST ORDER
- PRVNÍ, KTERÝ SE NELEŽDE ROZDĚL

2-APPROXIMATION

- JAKO FRACTIONAL, ALE ROZHODNOUT SE MEZI SOUČTEM VŠECH AŽ DO NEBO

DYNAMIC PROGRAMMING

- PSEUDOPOLYNOMIÁLNÍ ALGORITMUS
- NA BÁZI VYPLŇOVÁNÍ TABULKY

TSP

- NALÍT HAMILTONIAN CIRCUIT S NEJMENŠÍ VÁHOU V UNDIRECTED GRAFU K_n ($n \geq 3$)
- POKUD SE $|A,B| \neq |B,A| \rightarrow$ ASYMMETRIC
- 2 OPTIMIZATION PROBLEM
- ↳ BOUNDED BY POLYNOMIAL IN THE LENGTH OF INPUT
- ↳ IS CALLED STRONGLY NP-HARD IF P , ŽE L_P JE NP-HARD

NEJBLIŽŠÍ SOUSED - HEURISTIKA

DOUBLE TREE

- NALÍT MST
- ZDVŮSIT HRANY MST
- EULER TOUR V GRAFU, PŘESKAKOVAT JIŽ NAVŠTÍVENÉ
- 2-APPROXIMATION ALG.

CHRISTOFIDES

- 3-APPROXIMATION

HEURISTIKY PRO VYLEPŠENÍ ŘEŠENÍ

- NALÍT HAMILTONIAN CIRCUIT DE NEJAKÉ HEURISTIKY
- VYLEPŠIT POMOCÍ LOKÁLNÍ MODIFIKACE (NAPŘ. PŘECHÍT 2 HRANY)

4-OPT: DOUBLE BRIDGE

LOCAL SEARCH

ZÁVISÍ NA POČTU ČÍSEL POLYNOMIÁLNĚ
 NA JEJICH DĚLCI EXPONENCIÁLNĚ

SLOŽITOST $O(C \cdot n)$, LZE OHLEDNOUT K DĚLCI VSTUPU
 ALE NE VZHLÉDEM K VELIKOSTI VSTUPU

KOMU ROZVRHOVÁNÍ NA JEDNOM PROCESORU A NA PARALELNÍCH PROCESORECH. ROZVRHOVÁNÍ PROJEKTU S ČASOVÝMI OMEZENÍMI. PROGRAMOVÁNÍ S OMEZUJÍCÍMI PODMÍNKAMI.

ROZVRHOVÁNÍ = PŘÍRAZOVÁNÍ ^{ÚLOH} ÚKOLŮ ZDROJŮM V ČASE

GRAHAM'S NOTATION

RESOURCES | TASKS | CRITERION
 α | β | γ

SCHEDULING ON A RESOURCE

$1|t_j, d_j|C_{max}$ - NP-hard

- BRATLEY'S ALGORITHM

- BRANCH & BOUND

EVERY NODE LABELED:
- ORDER / COMPLETION OF LAST TASK

- BOUNDING WHEN DEADLINE EXCEEDED (ELIMINATE ALL BROTHERS ALSO)

$1|t_j| \sum C_j$ - NP-hard

• SHORTEST PROCESSING TIME FIRST

• EARLIEST DUE DATE FIRST

• EARLIEST DEADLINE FIRST

PARAMETRY
 t_j : RELEASE TIME
 p_j : PROCESSING TIME
 d_j : DUE DATE (SHOULD)
 $\tilde{d}_j$: DEADLINE (MUST)

PROHÉNNÉ

s_j : START TIME

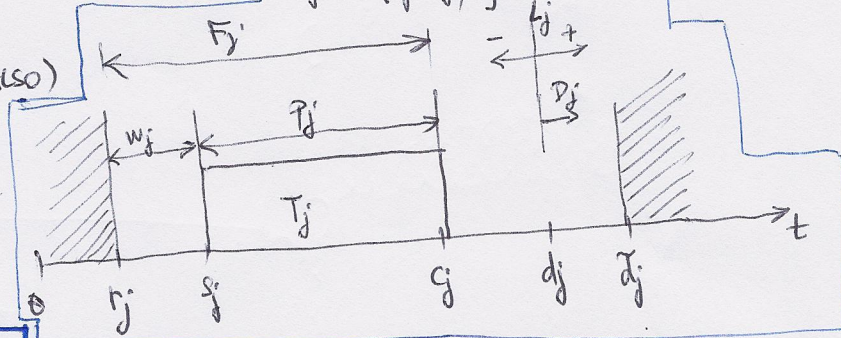
c_j : COMPLETION TIME

$w_j = s_j - t_j$: WAITING TIME

$F_j = c_j - t_j$: FLOW TIME

$L_j = c_j - d_j$: LATENESS

$D_j = \max\{c_j - \tilde{d}_j, 0\}$: TARDINESS



SCHEDULING ON PARALLEL IDENTICAL

• LONGEST PROCESSING TIME FIRST

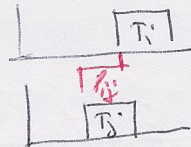
- LIST SCHEDULING

PROJECT SCHEDULING WITH TEMPORAL CONSTRAINTS

- JAKO DIRECTED GRAPH $(T_i \xrightarrow{t_{ij}} T_j)$

- PODMÍNKY: $s_i + t_{ij} \leq s_j$; $t_{ij} > 0$

$t_{ij} \leq 0$



PROGRAMOVÁNÍ S OMEZUJÍCÍMI PODMÍNKAMI: ILP PRO $PS1|temp|C_{max}$

TIME-INDEXED MODEL

+ JEDNODUŠE ROZŠÍŘITELNÝ NA PARALELNÍ IDENT. PROCESSORS

+ ILP NEPOTŘEBUJE MOC PODMÍNEK

+ VELIKOST MODELU RESTES UB

$x_{it} = 1$ iž $s_i = t$

RELATIVE-ORDER MODEL

+ NEZÁVISÍ NA UB

- MNOHO PODMÍNEK

$x_{ij} = 1$ iž $T_i < T_j$

OBA MODELY OBSAHUJÍ:

- PRECEDENCE CONSTRAINT

- RESOURCE CONSTRAINTS (PROTI PŘEKÝVÁNÍ ÚLOH)