Za ovakav ulaz mora da važi:

$$q(0,0,\ell_1) \wedge q(0,1,\ell_2) \wedge \dots \wedge q(0,s-1,\ell_s) \wedge \bigwedge_{k=s}^{n^d} q(0,k,\ell_0)$$

→ ovo je kompletan opis polazne konfiguracije

4) Završna konfiguracija:

$(\perp, a) \to \ell_i$ (tj. simbol koji karakteriše završnu konfiguraciju kodiramo sa ℓ_i)

U momentu n^d: $\bigvee_{i}\bigvee_{j=0}^{n^d} q(n^d, j, \ell') = 1$ (bar jedno da važi), pri čemu je ℓ' kod za $(\perp, a_k)$, $k=1,2,\dots,t$

↦ sviu ovim smo opisali rad naše mašine

Naša završna formula ϕ je konjukcija svih prethodnih formula.

Sada treba da odredimo složenost prethodnog (odredimo broj konjukcija i disjunkcija):

1) $\bigwedge_{i,j} \underbrace{\left(\bigvee_{\ell=1}^{t} q(i,j,\ell)\right)}_{\text{najviše } t \to \text{konstantno / ne utiče na složenost/}}$

n^{2d}

$\mapsto O(n^{2d})$

2) $\bigwedge_{\substack{i,j \\ \ell_1 \neq \ell_2}} \underbrace{(\neg q(i,j,\ell_1) \vee \neg q(i,j,\ell_2))}_{\text{dužina je 2}}$ → njih ima koliko ima parova t.d. je $\ell_1 \neq \ell_2$

n^{2d}

$t(t-1) \approx t^2$

$\mapsto O(n^{2d})$

2) Izdvojili NR-četvorke, četvorke zavise od Σ_1 i P

maximalno je $|NR| \leq t^4$ - izdvajanje možemo uraditi za konstantno vrijeme, jer ne zavisi od veličine ulaza.

$$\bigwedge_{i,j} \underbrace{[\neg q(\dots) \vee \neg q(\dots) \vee \neg q(\dots) \vee \neg q(\dots))}_{\text{veličina je 4}}$$

n^{2d} ← $(\ell_1, \dots, \ell_4) \in NR$

t^4

$\underbrace{t^4}_{\text{const}} \cdot (n^d-1)(n^d-1) = O(n^{2d})$

3) Kodiranje polazne konfiguracije - Linearno $\Rightarrow O(n^d)$

4) kodiramo završne konfiguracije

$$\bigvee_{\ell'} \bigvee_{j=0}^{n^d} q(n^d, j, \ell')$$

ℓ' - const.

$\Longrightarrow O(n^d)$

$\Longrightarrow$ Veličina naše formule se ponaša kao $O(n^{2d})$ - polinomijalna veličina.

Ovim je i dokaz teoreme završen ///

$\ulcorner$ $L \in NP$, $L_1 \in NPC$, $L_1 \subseteq L$ $\Big\}$ $L \in NPC$, ne važi obrnuto tj. $L, L_1 \in NP$, $L_1 \subseteq L$, $L \in NPC$ $\Big\}$ $\not\Longrightarrow L_1 \in NPC$

Zaključak: Ako u nekom jeziku imamo težak dio $\Rightarrow$ cijo jezik je težak, a obrnuto ne važi! $\lrcorner$

$\ulcorner$ 3SAT $\subseteq$ SAT

3SAT: Izdvajamo formule u kojima u svakoj klauzuli ima točno 3 literala, preciznije: $3SAT = \{\phi \mid \phi = \bigwedge D_i, D_i = x_{j_1}^{\delta_{j_1}} \vee x_{j_2}^{\delta_{j_2}} \vee x_{j_3}^{\delta_{j_3}} \wedge$ ϕ - zadovoljiva$\}$

??? $\phi \in 3SAT$

Postoji linearan algoritam koji provjerava da li je ϕ formula (tj. zadatak u KNF) //

$\lrcorner$

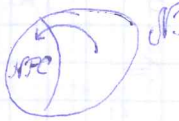
7. čas:

Na SAT možemo svesti sve zadatke za polinomno vrijeme. Dovoljno je da SAT svedemo na naš zadatak (3SAT), tada se i svaki zadatak može svesti na 3SAT. Ako SAT svedemo na X onda $X \in NPC$.

Teorema: $3SAT \in NPC$

dokaz: $D_i = \bigvee_{j=1}^{k} x_j^{\delta_j}$, D_i ujedinjavamo sa više klauzula D_i^{ℓ} t.d. $D_i = \bigwedge_{\ell} D_i^{\ell}$, pri čemu svako D_i^{ℓ} ima tačno tri člana.

Problem 3SAT je sličan problem SAT, ali time naređujemo da imamo tačno 3 elementa.



→ svaki problem iz NP za polinomno vrijeme možemo svesti na problem iz NPC.

$\Mapsto$ svaki problem iz NP se može svesti na 3SAT

1° slučaj: $k = 3$ $\quad D_i^1 = D_i$, $\ell = 1$

2° slučaj: $k > 3$ $\quad D_i = x_1^{\delta_1} \vee x_2^{\delta_2} \vee \dots \vee x_k^{\delta_k}$

Uvedimo novu promjenljivu y_1:

$D_i' \to (x_1^{\delta_1} \vee x_2^{\delta_2} \vee y_1) \wedge (\bar{y}_1 \vee x_3^{\delta_3} \vee \dots \vee x_k^{\delta_k})$

Ako je $D_i = 0$ (nije zadovoljivo) $\Rightarrow x_i^{\delta_i} = 0$ za $i = \overline{1,k}$, tada će nova klauzula biti $D_i' = y_1 \bar{y}_1 = 0$.

Ako je za neki izbor promjenljivih $D_i = 1 \Rightarrow \exists x_s^{\delta_s}$, $x_s^{\delta_s} = 1$

, ako je $s \le 2$ onda y_1 treba izabrati $y_1 = 0 \Rightarrow \bar{y}_1 = 1 \Rightarrow$

$x_1^{\delta_1} \vee x_2^{\delta_2} = 1$

, ako je $s > 2$ onda y_1 treba izabrati kao $y_1 = 1 \Rightarrow \bar{y}_1 = 0$

$\Rightarrow D_i \Leftrightarrow x_3^{\delta_3} \vee \dots \vee x_k^{\delta_k} = 1$

$\Mapsto$ nova formula zadovoljava.

Primjer kad uvedemo jednu promjenljivu tako da prva klauzula ima

3 člana, a druga više.

Sada uvodimo nove promjenljive $y_1, \dots, y_{k-3}$:

$$D_i \to \underbrace{(x_1^{\delta_1} \vee x_2^{\delta_2} \vee y_1) \wedge (\overline{y_1} \vee x_3^{\delta_3} \vee y_2) \wedge (\overline{y_2} \vee x_4^{\delta_4} \vee y_3) \wedge \dots \wedge (\overline{y_{k-4}} \vee x_{k-2}^{\delta_{k-2}} \vee y_{k-3})}_{Y_i} \wedge (\overline{y_{k-3}} \vee x_{k-1}^{\delta_{k-1}} \vee x_k^{\delta_k})$$

(ako je lijevo zadovoljivo, onda je i desno, i obratno)

$l = k-2$

($\Longrightarrow$): Ako je $D_i = 1$ (zadovoljiva), tada $\exists s : x_s^{\delta_s} = 1$

Kako izabrati y_i da Y_i bude zadovoljivo? Nova su klauzule oblika $(\overline{y_{s-2}} \vee x_s \vee y_{s-1})$. Ako izaberemo:

$$y_j = 1, \text{ za } j \le s-2 \quad i \quad y_j = 0, \text{ za } j > s-2$$

Mi tvrdimo da je tada formula zadovoljiva. Dokažimo

Sve disjunkcije oblika:

$$(\underbrace{a}_{x_i \text{ ili } \overline{y}} \vee x_{j+1}^{\delta_{j+1}} \vee y_j) = 1 \quad \text{za } j \le s-2$$

$$(\overline{y_j} \vee x_{j+2}^{\delta_{j+2}} \vee b) = 1 \quad \text{za } j > s-2$$

Jedino nam ostaje klauzula u kojoj učestvuje x_s, a jednostavno je vidjeti da je ona zadovoljiva.

$\Longrightarrow$ možemo izabrati y_i t.d. Y_i bude zadovoljiva

($\Longleftarrow$): ~~Kontrapozicijom~~

Ako je $D_i = 0 \Rightarrow x_i^{\delta_i} = 0$ onda je

$$Y_i = y_1 (\overline{y_1} \vee y_2)(\overline{y_2} \vee y_3) \dots (\overline{y_{k-2}} \vee y_{k-3}) \overline{y_{k-3}}$$

Pretpostavimo da je $Y_i = 1$

$y_1 = 1 \Rightarrow (\overline{y_1} \vee y_2) = (0 \vee y_2) = 1 \Rightarrow y_2 = 1$

$\vdots$

uopšte ako dođemo do $y_s = 1 \Rightarrow \overline{y_s} \vee y_{s+1} = 1 \Rightarrow y_{s+1} = 1 \quad \forall s \le k-4$

$\Longrightarrow y_s = 1$ za $\forall s = \overline{1, k-4}$

$\overline{y_{k-4}} \vee y_{k-3} = 1 \Rightarrow y_{k-3} = 1 \Rightarrow \overline{y_{k-3}} = 0$ a zbog oblika

f-je $\Rightarrow Y_i = 0$ (kontradikcija)

Ostalo je da provjerimo slučajeve za $k = 1$ i $k = 2$

3° slučaj: $k=2$

$D_i = x_1^{\delta_1} \vee x_2^{\delta_2}$ - ima dva člana, a nama treba tri pa dodajemo novu promjenljivu

$Y_i = (x_1^{\delta_1} \vee x_2^{\delta_2} \vee y_1)(\bar{y}_1 \vee x^{\delta_1} \vee x_2^{\delta_2})$

ako je $D_i = 0 \Leftrightarrow x_1^{\delta_1} = x_2^{\delta_2} = 0 \Rightarrow Y_i = y_1 \bar{y}_1 \equiv 0$

$D_i = 1 \Leftrightarrow \exists x_j^{\delta_j} = 1 \quad j \in \{1,2\}$

4° slučaj: $k=1$

$D_i = x_j^{\delta_j}$

uvodimo dvije promjenljive i D_i mijenjamo sa:

$D_i \leftrightarrow Y_i \equiv (x_j^{\delta_j} \vee y_1 \vee y_2)(x_j^{\delta_j} \vee y_1 \vee \bar{y}_2)(x_j^{\delta_j} \vee \bar{y}_1 \vee y_2)(x_j^{\delta_j} \vee \bar{y}_1 \vee \bar{y}_2)$

ako je $D_i = 0 \Leftrightarrow x_j^{\delta_j} = 0 \Rightarrow Y_i \equiv (y_1 \vee y_2)(y_1 \vee \bar{y}_2)(\bar{y}_1 \vee y_2)(\bar{y}_1 \vee \bar{y}_2)$

$\Rightarrow Y_i \equiv 0$

ako je $D_i = 1 \Leftrightarrow x_j^{\delta_j} = 1 \Rightarrow Y_i \equiv 1$

$\models$ Ova transformacija je složenosti $O(n)$, n-dužina formule

Obojivost grafa

Ako imamo graf $G=(V,E)$ (V - skup vrhova; E - skup grana) i skup boja $B = \{b_1, \dots, b_s\}$, onda preslikavanje $f: V \to B$ nazivamo bojenje grafa. $f(v_i)$ - boja čvora v_i

Bojenje grafa je pravilno ako dva susjedna čvora nemaju istu boju: $(\forall (u,v) \in E)\ f(u) \neq f(v)$

$|B| = s \to s$-broj boja

Ako je $|B| = k$, da li postoji $f: V \to B$ takvo da je bojenje pravilno? (Da li je graf G k-obojiv?) = zadatak k-obojivosti

$L = \{(G,k) \mid G$ se može obojiti sa k-boja$\}$ - jezik k-obojivosti

Zadatak k-obojivosti: Kad nam neko da G i k, treba provjeriti da li $(G,k) \in L$!

Ovo je bila formulacija zadatka o k-obojivosti.

Teorema: Problem k-obojivosti grafa je iz klase NPC.

dokaz ① Dokažimo da $L \in NP$.

Mjerilo veličine ulaza je broj čvorova grafa.

$v_1, v_2, \ldots$
$f(v_1), f(v_2), \ldots$ } ovaj izbor boja vršimo nedeterministički (izaberemo najbolje)

za $\forall (u,v) \in E$ provjerimo da li je $f(u) \neq f(v)$, ako je $f(u) \neq f(v)$ za $\forall (u,v) \in E$ → vratimo odgovor "DA", a ako $\exists (u,v)$ t. d. je $f(u) = f(v)$ → vratimo odgovor "ne".

$E \subseteq V \times V \Rightarrow |E| \leq |V|^2$

Izbor boja košta koliko imamo čvorova, pa je to linearno vrijeme.

② Dokažimo da $L \in NPT$.

Sada svedimo jezike na L, tj. dokažimo da je $L \in NPT$. Dovoljno je jedan NP problem da svedemo na L. Uzmimo da 3SAT svedemo na L za polinomijalno vrijeme.

Neka su naše promjenljive $x_1, \ldots, x_m$ (u formuli $\phi = \bigwedge_{i=1}^{k} D_i$).

Mi ćemo formirati graf sa čvorovima (V): $x_1, \ldots, x_n, \bar{x}_1, \ldots, \bar{x}_n$, dodaćemo čvorove $y_1, \ldots, y_n$ i za svaku klauzulu uvesti po jedan čvor $D_1, \ldots, D_k$. Skup grana tog grafa (E) opisaćemo na sljedeći način:

$(x_i, \bar{x}_i)$ $\forall i \in \{1, \ldots, n\}$

(y_i, y_j) za $\forall i \neq j$, $i, j \in \{1, \ldots, n\}$

(x_i, y_j) za $\forall i \neq j$ $\quad (\bar{x}_i, y_j)$ za $\forall i \neq j$

D_i je tačna ako je bar jedna promjenljiva unutar nje $= 1$

$(x_i^{\delta_i}, D_j)$, ako $x_i^{\delta_i}$ ne učestvuje u D_j.

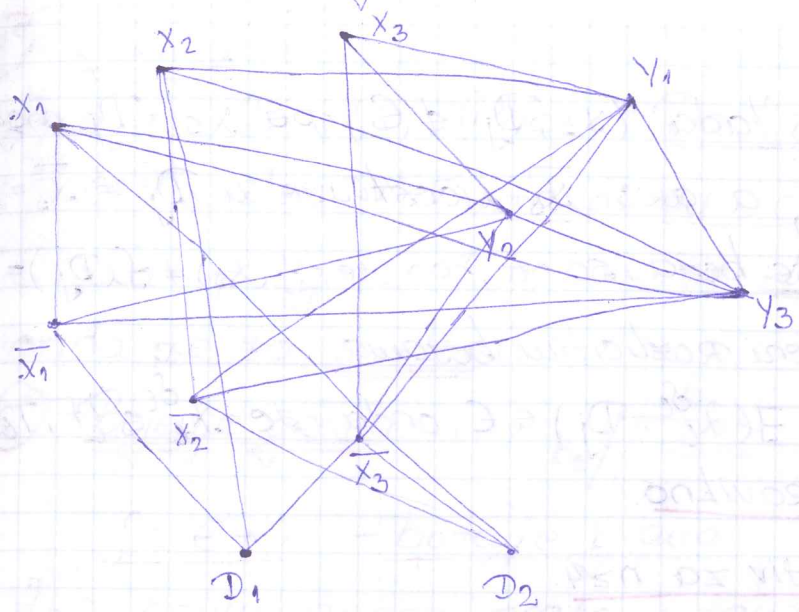
→ ovo je algoritam kojim se formira novi graf $G = (V, E)$ i pitamo se da li se može obojiti (pravilno) sa $n+1$-bojom?

Konkretan primjer: x_1, x_2, x_3

$$\phi = (\underbrace{x_1 \vee \bar{x}_2 \vee x_3}_{D_1})(\underbrace{\bar{x}_1 \vee x_2 \vee x_3}_{D_2})$$

V: $x_1, x_2, x_3, \bar{x}_1, \bar{x}_2, \bar{x}_3, y_1, y_2, y_3, D_1, D_2$

Sada formiramo graf:



Pitanje: Da li je moguće ovaj graf obojiti sa 4-boje?

To je moguće $\Leftrightarrow$ ϕ zadovoljiva

Teorema: G je obojiv sa $n+1$-bojom $\Leftrightarrow$ ϕ zadovoljiva

dokaz:

Definišemo skup boja $B=\{T_1,\dots,T_n,F\}$, gdje T_i igra ulogu 1, a F igra ulogu 0.

($\Leftarrow$): Neka je ϕ zadovoljiva $\Rightarrow \exists(d_1,\dots,d_n)$ t.d. je $\phi(d_1,\dots,d_n)=1$

Ako je $d_i=1$ onda x_i boji sa T_i, a $\bar{x}_i$ boji sa F. Svako y_i boji sa T_i.

Ako je $d_i=0$ onda x_i boji sa F, a $\bar{x}_i$ sa T_i.

Kako obojiti D_i?

Kako je $D_i=1$ (jer je ϕ zadovoljiva) tada $\exists\, x_j^{\delta_j}$ tako da je $x_j^{\delta_j}=1$ i $x_j^{\delta_j}$ učestvuje u formiranju D_i. Onda D_i boji sa bojom od $x_j^{\delta_j}$ (bojom literala koji je tačan)

Tvrdimo da je ovo bojenje pravilno. Uzmimo dva susjedna čvora i dokažimo da su obojeni drugim bojama.

To se lako pokazuje - očigledno je, sem za $(x_i^{\delta_i}, D_j)$.

$(x_i^{\delta_i}, D_j)\in E$ proizvoljno (pri čemu $x_i^{\delta_i}$ ne učestvuje u D_j)

D_j $x_s^{\delta_s}=1 \wedge x_s^{\delta_s}$ učestvuje u D_j ; $D_j \to T_s$

x_s i $\bar{x}_s$ su jedini interesantni [jer ~~je~~ $x_i = F \vee x_j = T_j \neq T_s = x_s$ za ~~$j \neq s \Rightarrow f(x_i) \neq f(x_s^{ds})$~~]

Ako x_s učestvuje u D_j tada $(x_s, D_j) \notin E$, pa x_s i D_j mogu biti obojeni istom bojom, a ako $\bar{x}_s$ učestvuje u $D_j \Rightarrow \bar{x}_s = 1$ $\Rightarrow x_s = 0 \Rightarrow l_s = 0 \Rightarrow x_s$ se boja sa F, pa je $f(x_s) \neq f(D_j) = T_s$.

Dakle, x_s i $\bar{x}_s$ su obojeni različitim bojama.

Neka je x_ℓ^{ds}, $\ell \neq s$. Ako $\exists (x_\ell^{ds}, D_j) \in E$ onda je $x_\ell^{ds} \in \{F, T_\ell\}$

$\Longleftarrow$ ovo bojenje je pravilno.

($\Longrightarrow$): Graf G je obojiv za $n \geq 4$.

[Koristimo kontrapoziciju tj. pretpostavimo da je $\phi = 0$ (da nije zadovoljiva). ~~X~~ ne treba!

Za $\forall i$ (pošto su svi povezani) koristimo n-boja. Najmanje koristimo $n+1$ boja.]

Neka je G graf $(n+1)$ obojiv. Dokažimo da je ϕ zadovoljiva.

Ako je x_i obojiva sa T_i, tada x_i dodijelimo 1, a ako je obojena sa F, $x_i = 0$.

Dokažimo da D_i nije obojeno sa F, ako je $n \geq 4$. Neka je D_i obojeno bojom T_s. To znači da D_i nije povezano sa x_s ili sa $\bar{x}_s$. Ako je x_s obojeno sa T_s, onda x_s učestvuje u D_i pa je $D_i = 1$ (jer je $x_s = 1$). A ako je $\bar{x}_s$ obojeno sa T_s, tada $\bar{x}_s$ učestvuje u D_i i $\bar{x}_s = 1$ (jer je x_s tada obojeno sa F) $\Rightarrow x_s = 0 \Rightarrow \bar{x}_s = 1) \Longrightarrow D_i = 1$

Svaka klauzula je po ovakvom dodjeljivanju $= 1$, pa je $\phi = 1$ (ϕ je zadovoljiva). ///

Složenost algoritma:

Samo formiranje čvorova je linearno. Za povezivanje D_i troši se najviše kvadratno vrijeme. Tako da je algoritam za ovu transformaciju polinomijalan $\Rightarrow$ NP-C zadatak.

Primjeri NPC zadataka:

1° K-klika:

Ako je $G = (V, E)$. Pitanje: Da li postoji podgraf $G' = (V', E')$ ($V' \subseteq V \wedge E' \subseteq E \cap V' \times V'$) takav da je $E' = V' \times V'$ tj. da je kompletan? Ovakav podgraf G' naziva se klika, a ako je $|V'| = k$ onda je to k-klika.

2° Bojenje grana

$f: E \to B$ – bojenje grana

Ako bojimo grane grafa i tada možemo govoriti o pravilnom bojenju. Grane su pravilno bojene ako su susjedne grane različitih boja tj. $f(u,v) \neq f(v,w)$

Sa koliko boja možemo obojiti grane?

Broj grana koje počinju (završavaju se) u čvoru je STEPEN ČVORA, a STEPEN GRAFA je maximalni stepen čvora u njemu. Ako je stepen grafa m, onda nam za bojenje grafa treba najmanje m-boja.

Teorema: Ako je stepen grafa m, onda se grane grafa mogu pravilno obojiti sa $m+1$ bojom.

Pitanje: Da li se grane grafa G mogu obojiti sa m-boja, pri čemu je m-stepen grafa? Ovaj zadatak je NPC.

3° Problem trgovačkog putnika

Dat je graf. Da li možemo naći put da svaki čvor obiđemo tačno jednom i vratimo se u polazni čvor?

P, NP, NP-težak, NPC ... Da li postoje još neke klase zadataka? Da, to su:

PSPACE – (zadaci) jezici L koji su polinomijalni po prostoru, tj. $\exists$ DTM M i postoji polinom $p(*)$ t.d. $S(M,n) \leq p(n)$ i $L_M = L$.

$= \{L : \exists DTM\ M, \exists p(x) : S(M,n) \leq p(n) \wedge L_M = L\}$

Ako uzmemo NDTM umjesto DTM u prethodnoj definiciji dobijamo klasu NPSPACE

PSPACE = NPSPACE (1970. Savič)

$P \subseteq NP \subseteq PSPACE = NPSPACE$

EXPTIME - klasa jezika koji mogu biti prepoznata za deterministički eksponencijalno vrijeme.

NEXPTIME - - " - nedeterministički - " -

$P \subseteq NP \subseteq PSPACE = NPSPACE \subseteq EXPTIME \subseteq NEXPTIME \subseteq DEXPTIME$ ($\downarrow 2^{2^n}$)

$P \subseteq NP \subseteq PSPACE \subseteq EXPTIME \subseteq NEXPTIME \subseteq EXPSPACE \subseteq DEXPTIME$

NPC ⊆ NP (najteži zadaci iz NP su NPC)

Trenutno se zna: $P \neq EXPTIME$, ostalo se ne zna.

C - proizvoljna klasa jezika

C, C-težak, C=kompletan

Ako je C deterministička klasa $\Rightarrow C = C$-kompletan

C_0-C - komplement klase C

Mi smo pitali da li $x \in L$, a možemo postaviti pitanje: Da li $x \notin L$, tj. da li $x \in L^C$?

To je jedino interesantno kod nedeterminističkih klasa (Npr. C_0-NP)

Tačan odnos između NP i C_0-NP se ne zna

Primjer jezika koji pripada klasi PSPACE je jezik:

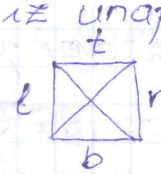
Razmatrali smo da li $\exists(x_1, \dots, x_n)$ t.d. je Bulova f-ja $\phi(x_1, \dots, x_n) = 1$. A C_0-klasa bila bi: $\forall x_1, \dots, \forall x_n\ \phi(x_1, \dots, x_n) = 0$

Sledeći problem: $Q_1 x_1 Q_2 x_2 \dots Q_n x_n\ \phi(x_1, \dots, x_n) = 1$, $Q_i \in \{\exists, \forall\}$ pripada PSPACE-kompletnim zadacima (najtežim u klasi PSPACE)

8. čas:

Problem popločavanja

Def. *Ploča* je kvadrat dimenzije 1×1 čije su ivice obojene sa bojama iz unapred fiksiranog skupa boja.



umjesto bojanja ivica – bojanje 4 trougla

Matematička formulacija:

Fiksiramo skup boja B. Ploča je uređena četvorka: $(l, b, r, t) \in B^4$. Ploča ne dozvoljava rotaciju i simetriju $\Rightarrow$ važan red u četvorki (uređena četvorka).

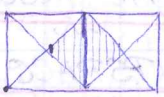
Sa T označavamo skup (konačan) različitih ploča.

Opšti problem popločavanja:

Dat je dio ravni izdijeljen na kvadrate dimenzije 1×1 i pitanje je da li se taj dio ravni može pravilno popločati sa pločama iz ravni T, pri čemu je dozvoljeno da se neka ploča ponavlja više puta.

T - čuva tip ploče

pravilno popločati = susjedne ploče koje imaju zajedničku ivicu, trouglovi koji odgovaraju toj ivici su iste boje.



Imamo više problema za popločavanje:

1° *$n \times n$ popločavanje:*

Ako je dat kvadrat $n \times n$ čije su ivice obojene bijelom bojom, da li se on može pravilno popločati sa pločama iz T.

2° *popločavanje koridora:*

Dat je broj n, da li postoji m tako da se pravougaonik dimenzije $n \times m$ može pravilno popločati sa pločama iz sk. T.

(sve ivice su bijele boje)

3° n×n popločavanje sa dva igrača:

Dat je pravougaonik dimenzije n×n čije su sve ivice bijele boje i dva igrača A i B koji naizmenično biraju tip ploče i postavljaju na sledeće polje. Igrač A se trudi da popločа ovaj pravougaonik, a B ga spečava u tome. Pod predpostavkom da A i B imaju najbolju strategiju, pitanje je da li A može da pobjedi, da popločа ovaj pravougaonik?

Sva tri problema su odlučiva i daju odgovor "da" ili "ne". Postoje i drugi problemi koji nisu odlučivi.

Nas interesuje složenost:

Teorema:

a) ako je n zadato unarno onda problemi imaju sledeću složenost:

- n×n popločavanje je NP-kompletno.
- koridor popločavanje je PSPACE-kompletno
- n×n popločavanje sa dva igrača je EXPTIME-kompletno.

b) ako je n zadato binarno onda prethodni problemi pripadaju sledećim klasama:

- n×n popločavanje je NEXPTIME-kompletno.
- popločavanje koridora je EXPSPACE-kompletni
- n×n popločavanje sa dva igrača je DEXPTIME-kompletno (dvostruko eksponenc. kompl.)

Primjer: n = 1000

$(\underbrace{1 \cdot 1 \cdot 1 \cdot \ldots \cdot 1}_{\times 1000})$ - unarno = ulaz veličine 1000

binarno = ulaz veličine log 1000

dokaz: a) n×n popločavanje

Da bi dokazali da je ovaj problem NP-C, treba prvo pokazati

da je NP.

Neka je dato τ (ne zavisi od veličine ulaza) i pravougaonik dimenzije $n \times n$.

Pitanje: Da li pravougaonik možemo ili nemožemo popločati?

Nedeterministički izaberemo ploču po ploču i rečemo ih - za ovu radnju nam je potrebno $O(n^2)$ vremena.

Koliko ima ploča, 4 puta više ima ivica, koje mi treba da provjeravamo da li su dobro obojene - a za to je dovoljno $O(n^2)$ vremena (n^2-ivica koje treba da provjerimo).

Dovoljno je da bude odgovor "nije" ako nađe na jednu ivicu koja nije dobro obojena.

$L^{\tau}_{n \times n} \overset{def}{=} \{ L : L$ je kvadrat $n \times n$ koji možemo pravilno popločati$\}$

$L^{\tau}_{n \times n} \in NP$ ✓

Kad nam neko da kvadrat, pitamo se da li on pripada jeziku.

Drugi dio dokaza: Ovaj problem (jezik) je NP-težak. → ovo treba dokazati!

≡ Proizvoljan jezik iz NP je svodljiv na ovaj problem.

Koristićemo Tjutingovu mašinu da pokažemo to.

Neka je L_1 proizvoljan jezik $\in NP$. Dokažimo da L_1 možemo za polinomno vrijeme svesti na $L^{\tau}_{n \times n}$ (tj. L_1 nije teži od $L^{\tau}_{n \times n}$)

Kako je $L_1 \in NP \Rightarrow \exists$ NDTM M i $\exists$ polinom $p(\,)$ tako da je složenost $T(M, n) \leq p(n)$ i da je jezik $L_M = L_1$.

Poznato je iz algebre da $\exists d$ t.d. $\forall n \in \mathbb{N}$ $p(n) \leq n^d$.

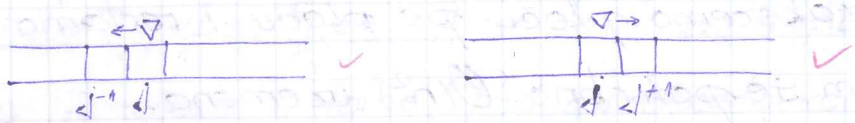
Drugim riječima, $T(M, n) \leq n^d$ za $n \neq 1$.

Čim postoji ovakva mašina, ona će najviše preći na traci n^d polja.

Izvršićemo normalizaciju mašine:

U mašini imamo neke petorke $(\_, \_, q', \_, L)$ i $(\_, \_, q', \_, R)$,

ovakve petorke neće odgovarati mašini, pa ćemo q' zamijeniti sa q'_L i q'_R. Znači prethodne petorke mijenjamo sa sledećim petorkama: $(-,-,q'_L,-,L)$ i $(-,-,q'_R,-,R)$ - ovo su dva ekvivalentna stanja.



Takođe uvodimo pretpostavku da mašina ne ide na negativna polja.

Broj stanja se povećа duplo - za duplo se uvećа i veličina programa.

Kako opisati samo kodiranje, svođenje jezika na popločavanje?

Kodiramo: 1° ulaza (polazne konfiguracije)

2° Rada mašine (pravilnog prelaženja iz jedne u drugu konfiguraciju)

3° završne konfiguracije

Drugi korak (2°) rešavamo na sledeći način:

Za naše ploče izabraćemo posebne boje za horizontalne i posebne za vertikalne ivice.

$\{b\}$ (b - bijela) $\cup Q$ - za vertikalne ivice

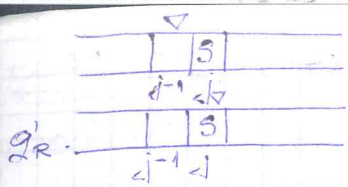
$\Sigma \cup (Q \times \Sigma)$ - za horizontalne ivice

Imamo sledeće tipove ploča:

1° Neka je: C_i | s |
C_{i+1} | s |

glava nije u okolini tog polja - simbol se prepisuje. Tada uvodimo tip ploče

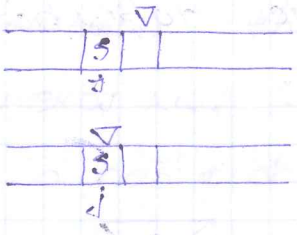
2° Ako imamo da je glava iznad nekog polja i pomjera se desno na naše polje. Da bi imali poziciju glave mi ovo kodiramo sa (q'_R, s)



i uvodimo tip ploče:

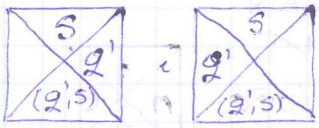
(glava dolazi sa desne strane)

3° Treći tip ploče: pomjeranje glave ulijevo tj.



Uvodimo sledeći tip ploče:

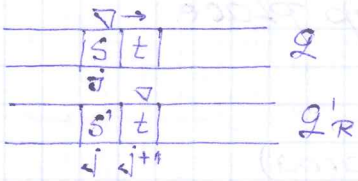
Napomena: Uvijek smo normalizaciju, jer ako bi imali



i ako ih spojimo – ispašće da ima dvije glave.

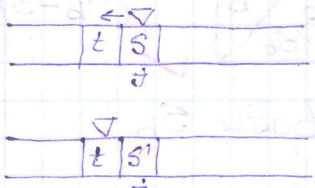
4° Četvrti tip ploče:

Imamo sledeću situaciju



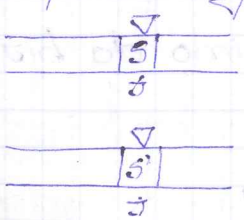
tj. imamo u programu petorku (q, s, q'_R, s', R), pa uvodimo sledeći tip ploče:

5° Peti tip ploče: (isto kao 4° samo pomjeranje ulijevo)



tj. imamo petorku (q, s, q'_L, s', L), pa uvodimo tip ploče:

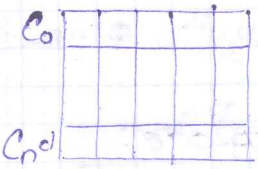
6° Šesti tip: ostaje glava iznad istog polja



tj. imamo petorku $(q, s, q', s', 0)$, onda uvodimo ploču

→ ovim tipovima ploča opisujemo pravilne prelaze iz jedne konfiguracije u drugu.

Kako kodirati pravilno ulaz i završno stanje?



Polaznu konfiguraciju simuliramo tako što bojimo tu polaznu ivicu datim bojama ili uvođenjem novih ploča:

ulaz je $S_0 S_1 \dots S_n$.

za S_2 bi uveli ploču

. . .

u nekom momentu n neki specijalan tip ploče

i dalje

- simbol praznog polja

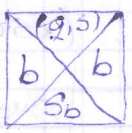
Na većem dijelu ulaza je posljednji tip ploče.

Slično za završnu konfiguraciju:

(bojenje zadnje linije specijalnim bojama)

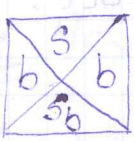
Mi možemo nametnuti posebne ploče koje nas dovode do bijele boje. $(q, S, \perp, S', M)$

Uvodimo specijalne simbole



(S_b - specijalna boja)

b nam omogućava da širimo boju S_b.



(b - specijalna boja (ne bijela)) — na ovaj način sve obojamo sa S_b.

a imamo i mogućnost da S_b prevedemo u bijelu boju:

i na kraju

- dolazimo do bijele boje

Koliko nam treba ovakvih ploča?

Konačan broj ploča i za to kodiranje nam je potreban

polinomno vrijeme (jedino kodiranje ulaza može da napravi produženje vremena)

Uzimamo pravougaonik: $n^d \times n^d$

Ulaz prihvaćen $\Leftrightarrow$ postoji pravilno popločavanje

Tražimo efikasnu tehniku da uvidimo da li neki problem pripada nekoj klasi: dovoljno je da problem popločavanja svedemo na naš problem.

9. čas:

Algoritmi u teoriji brojeva

Neka su $m,n \in \mathbb{Z}^+$. Nas interesuje da n zapišemo kao $n = q \cdot m + r$, $0 \le r < m$, gdje je r - ostatak, a $q = \left[\frac{n}{m}\right]$ - količnik.
Ova reprezentacija je jedinstvena.
Ako je $r = 0$ onda kažemo da je n djeljivo sa m ili da m dijeli n. Oznaka je m/n.

Definicija: Za broj $p \ge 1$ kažemo da je prost ako za $\forall k$ iz uslova $k/p \Rightarrow k = 1 \vee k = p$

Teorema: (Osnovna teorema aritmetike)
Za svaki $n \in \mathbb{Z}^+$ možemo zapisati u obliku $n = p_1^{a_1} \cdot \ldots \cdot p_l^{a_l} = \prod_{i=1}^{l} p_i^{a_i}$ (*) gdje su p_i međusobno različiti prosti brojevi.
- Ako su $a_i \neq 0$ onda je ova reprezentacija jedinstvena.

Def. Broj djelilaca broja n označavaćemo sa $d(n)$, drugim riječima $d(n)$ je kardinalnost skupa čiji su elementi svi djelioci broja n koji su manji ili jednaki od n.

(*) - kanonska reprezentacija

$$A(n) = \{k \in \mathbb{Z}^+ : k/n\}$$
$$d(n) = |A(n)|$$

$k/n \Rightarrow k/p_1^{a_1} \cdot \ldots \cdot p_l^{a_l} \Rightarrow k = p_1^{\alpha_1} \cdot \ldots \cdot p_l^{\alpha_l}$, $0 \le \alpha_i \le a_i$
Ovakvih brojeva ima koliko ima i l-torki $(\alpha_1, \ldots, \alpha_l)$ koje zadovoljavaju prethodne uslove.

$$k \longmapsto (\alpha_1, \ldots, \alpha_l)$$
(jednoznačno; α_1: $0, \ldots, a_1$; α_l: $0, \ldots, a_l$)

$$d(n) = (a_1+1)(a_2+1)\cdots(a_l+1)$$

Primjer: $n = 18 = 2^1 \cdot 3^2$

$d(18) = (1+1)(2+1) = 6 \Rightarrow$ ima 6 djelioca broja 18. i to su brojevi oblika $k = 2^{\alpha_1} 3^{\alpha_2}$, $0 \le \alpha_1 \le 1$, $0 \le \alpha_2 \le 2$.

$\alpha_1 = 0 \quad \alpha_2 = 0 \Rightarrow k = 1$

$\alpha_1 = 1 \quad \alpha_2 = 0 \Rightarrow k = 2$

$\alpha_1 = 0 \quad \alpha_2 = 1 \Rightarrow k = 3$

$\alpha_1 = 1 \quad \alpha_2 = 1 \Rightarrow k = 6$

$\alpha_1 = 0 \quad \alpha_2 = 2 \Rightarrow k = 9$

$\alpha_1 = 1 \quad \alpha_2 = 2 \Rightarrow k = 18$

Def.: Neka su $m, n \in \mathbb{Z}^+$. Broj $g \in \mathbb{Z}^+$ nazivamo NZD(m,n) ako važi:

1° $g \mid n \wedge g \mid m$

2° $(\forall t)\,(t \mid n \wedge t \mid m \Rightarrow t \mid g)$

- umjesto NZD(n,m) pisaćemo samo (n,m)

Def.: Za brojeve $m, n \in \mathbb{Z}^+$ kažemo da su uzajamno prosti ako važi da je NZD$(m,n) = 1$.

Primjer: brojevi 4 i 9 su uzajamno prosti

Def.: Broj pozitivnih cijelih brojeva koji su manji od n a uzajamno su prosti sa n označavamo sa $\gamma(n)$. $\gamma(n)$ se naziva Ojlerovom funkcijom

$$B(n) = \{m \in \mathbb{Z}^+ \mid m < n \wedge \text{NZD}(n,m) = 1\}$$

$$\gamma(n) = |B(n)|$$

Ako znamo kanonsku reprezentaciju našeg broja, kako da izrazimo $\gamma(n)$?

$$n = p_1^{a_1} \cdot \ldots \cdot p_t^{a_t} \Rightarrow \gamma(n) = ?$$

1° $n = p$ (n je prost broj)

$$\gamma(p) = p - 1$$

2° n nije prost

$$\gamma(n) = n - |\{m : \text{NZD}(m,n) > 1\}|$$

→ od n oduzimamo kardinalnost skupa čiji su elementi oni koji nisu uzajamno prosti sa n.

$= n - \underbrace{\frac{n}{p_1}}_{\text{oni koji su djeljivi sa } p_1 \text{ a manji su od } n} - \frac{n}{p_2} - \dots - \frac{n}{p_t} - ? \dots ?$

$x : p_1 \mid x \wedge x < n$

a njih je: $1 \cdot p_1, 2 \cdot p_1, \dots, \frac{n}{p_1} \cdot p_1$

- u ovom oduzimanju smo oduzeli sve parove oblika $p_1 p_2$ dva puta (npr. $2 \cdot 3 \wedge 3 \cdot 2$)
- a x koji su djeljivi sa $p_1 p_2$ tj. i sa p_1 i sa p_2:

$1 \cdot p_1 p_2, 2 p_1 p_2, \dots, \frac{n}{p_1 p_2} \cdot p_1 \Rightarrow$ ima ih $\frac{n}{p_1 p_2}$

pa je formula:

$$\varphi(n) = n - \frac{n}{p_1} - \dots - \frac{n}{p_t} + \underbrace{\frac{n}{p_1 p_2} + \dots + \frac{n}{p_{t-1} p_t}}_{\text{vratili smo neke duplo}} - \frac{n}{p_1 p_2 p_3} - \frac{n}{p_1 p_2 p_4} - \dots - \frac{n}{p_{t-2} p_{t-1} p_t}$$

$$+ \dots + (-1)^t \frac{n}{p_1 \dots p_t} \quad \lceil t \text{ parno} \to "+",\ t \text{ neparno} \to "-" \rfloor$$

$$= n\left(1 - \frac{1}{p_1}\right)\left(1 - \frac{1}{p_2}\right) \dots \left(1 - \frac{1}{p_t}\right)$$

Euklidski algoritam

Zapišimo n kao $n = q \cdot m + r$, $q = \left\lfloor \frac{n}{m} \right\rfloor$, $r = n \text{ MOD } m$

$q = NZD(n, m)$

$n = q \cdot m + r \Rightarrow r = n - qm$

Pošto $q \mid n \wedge q \mid m \Rightarrow q \mid r$. Ako bi $q \mid r \wedge q \mid m \Rightarrow q \mid n$

Pa zbog prethodnog: $q = NZD(n, m) = NZD(m, r) = NZD(m, n \bmod m)$

Ovim smo opisali Euklidski algoritam za traženje NZD.

$NZD(n, 0) = n$ - algoritam se zaustavlja

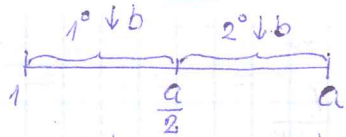
Algoritam:

```
function NZD(n,m);
    begin
        if m=0 then NZD:=n;
        else
            NZD:= NZD(m, n MOD m);
    end;
```

Složenost ovakvog algoritma je:

Lema: Ako je $1 \leq b \leq a$ onda je $a \bmod b \leq \frac{a-1}{2}$.

dokaz:



1°) $b-1 \leq a-b \iff 2b \leq a+1 \iff b \leq \frac{a+1}{2}$

$a = qb + r, \quad 0 \leq r < b \qquad r = a \bmod b$

$\Downarrow$

$r \leq b-1 \Rightarrow a \bmod b \leq b-1 \leq \frac{a+1}{2} - 1 = \frac{a-1}{2}$

2°) $b-1 > a-b \iff 2b > a+1 \iff b > \frac{a+1}{2} \iff -b < -\frac{a+1}{2}$

$a = qb + r \Rightarrow r = a - qb$

$b \leq a \Rightarrow q = \left\lfloor \frac{a}{b} \right\rfloor \geq 1 \iff -q \leq -1$

$r = a - qb \leq a - b$

$a \bmod b \leq a - b < a - \frac{a+1}{2} = \frac{a-1}{2}$

Teorema: Ako su data dva broja $a, b \in \mathbb{Z}^+$ onda Euklidov algoritam nalazi $NZD(a,b)$ za najviše $\lfloor 2\log_2 M \rfloor + 1$ dijeljenja gdje je $M = \max\{a,b\}$.

dokaz:

Pretpostavimo da je $a \geq b$ tj. $M = a$.

U prvom koraku formiramo: $a_0 = a$, $a_1 = b$, $a_2 = a \bmod b = a_0 \bmod a_1$. Na osnovu a_0 i a_1 formiramo a_2, na osnovu a_1 i a_2 formiramo a_3, $a_3 = a_1 \bmod a_2$ i u k-tom koraku formiramo $a_k = a_{k-2} \bmod a_{k-1}$. Ovaj algoritam radi dok a_k ne postane nula tj. dok jedan od brojeva a_{k-2} i a_{k-1} iz prethodne formule ne postane nula.

$a_{i-2} \quad a_{i-1} \quad a_i \xrightarrow{\text{Lema}} a_i \leq \frac{a_{i-2}-1}{2}$

$\Rightarrow \quad a_2 \leq \frac{a_0 - 1}{2} \qquad a_3 \leq \frac{a_1 - 1}{2}$

$a_2 \leq \frac{M-1}{2} \qquad a_3 \leq \frac{M-1}{2}$

$a_4 \leq \frac{a_2 - 1}{2} \qquad a_5 \leq \frac{a_3 - 1}{2}$

$\vdots \qquad \vdots$

Ako iskoristimo strogu nejednakost imamo:

$a_2 < \frac{M}{2} \qquad a_3 < \frac{M}{2}$

$a_4 < M/2^2 \qquad a_5 < M/2^2$

$a_6 < \frac{M}{2^3}$ $\quad a_7 < \frac{M}{2^3}$

opštije:

$$\left.\begin{array}{l} a_{2i+1} < \frac{M}{2^i} \\ a_{2i} < \frac{M}{2^i} \end{array}\right\} \Rightarrow a_k < \frac{M}{2^{\lfloor k/2 \rfloor}}$$

Mi očekujemo da se algoritam završi kad je $a_k = 0$.

Koliko nam za to treba koraka?

$a_i \in \mathbb{Z}^+$. Ako izaberemo takvo k da je $\frac{M}{2^{\lfloor k/2 \rfloor}} < 1$ (tada će $a_k < 1 \Leftrightarrow a_k = 0$.

$$M < 2^{\lfloor k/2 \rfloor} \Leftrightarrow \lfloor \tfrac{k}{2} \rfloor > \log_2 M \Leftrightarrow k > \lfloor 2\log_2 M \rfloor$$

- A ako bi nam pretpostavka bila $b > a$, ponovili-bismo isti postupak samo bi zamijenili mjesta a i b. //

Broj koraka je $\leq \lfloor 2\log M \rfloor + 1$

- kako se ovo odnosi na alg. Euklidov?

Ulaz a i b. Veći od njih označimo sa M. Za zapisivanje a i b treba nam $O(\log M)$ bita (veličina ulaza).

Složenost dijeljenja za dati ulaz je $O(\log^2 M)$. Pošto nam treba da ponovimo $O(\log M)$ koraka $\Rightarrow$ Ukupno naš košta $O(\log^3 M)$ - polinomijalno od veličine ulaza

Moguć je bolji algoritam - $O(\log M)$ (ideja: ako su a i b parni brojevi, onda im se skidaju zajedničke dvojke tj. $a = 2^k a_1$ i $b = 2^k b_1$ i posmatramo samo a_1 i b_1)

10. čas:

Prošireni Euklidov algoritam

$m, n \in \mathbb{Z}^+ \quad g = NZD(m,n)$

Napravimo linearnu kombinaciju brojeva m i n:

$$z = t \cdot n + u \cdot m, \quad t, u \in \mathbb{Z}, \quad 0 \neq z \in \mathbb{Z} \qquad (1)$$

Ne gubeći na općenitosti uzećemo da je $n > m$.

Smatraćemo da je jednačina (1) - jednačina po nepoznatima t, u koje se traže u skupu $\mathbb{Z}$.

Pitanje: Da li postoji rješenje (t,u) tj. postoji li linearna kombinacija datih brojeva m i n koja daje datu vrijednost z?

Prošireni Euklidov algoritam:

1° daje potvrdan ili odričan odgovor na pitanje postoji li rješenje jednačine (1)

2° ako je odgovor potvrdan onda još i saopštava r-nje (t,u).

3° saopštava vrijednost $NZD(a,b)$

* Naziv "proširen" je jer pored računanja NZD (što radi obični Euklidov algoritam), on traži i brojeve t i u.

Ako neki broj dijeli n i m onda on dijeli i z tj.

$$g \mid n \wedge g \mid m \Rightarrow g \mid z$$

Obrnuto ne mora da važi!

Od prethodnog časa znamo: $NZD(m,n) = NZD(n \bmod m, m)$.

$$g = t \cdot n + u \cdot m = t'(n \bmod m) + u' \cdot m$$

Postavljamo pitanje koja je veza između t i u sa t' i u'.

$$g = t'(n \bmod m) + u' \cdot m = t'\left(n - \left\lfloor \frac{n}{m} \right\rfloor \cdot m\right) + u' \cdot m$$

$$= t' \cdot n + \left(u' - t' \left\lfloor \frac{n}{m} \right\rfloor\right) \cdot m$$

$$\Longrightarrow \underline{t = t'} \wedge \underline{u = u' - t' \left\lfloor \frac{n}{m} \right\rfloor}$$

Zaključak: ako postoje cijeli koeficijenti $(t$ i $u)$ za Ln kombinaciju brojeva n i m onda postoje i koeficijenti $(t'$ i $u')$ za Ln.

kombinaciju manjeg para brojeva $(n \bmod m$ i $m)$, vrijedi i obrnuto, pošto se t i u mogu izraziti preko t' i u'.

Dakle, mi ćemo iz koraka u korak svoditi pitanje na pitanje istog oblika samo za par brojeva koji je sve manji i manji. Sve dok ne svedemo pitanje na par $(0, m)$ tj. $NZD(0,m) = m = q$. Tada možemo napisati $0 \cdot 0 + 1 \cdot m = m \Rightarrow t = 0$ i $u = 1$. Zatim u drugoj fazi algoritma krećemo u suprotnom smjeru tj. na osnovu t' i u' računamo t i u koji odgovaraju većem paru.

Cijeli je izložen ovaj algoritam.

```
NZDEXT(m, n ; q, t, u)
        ulaz      izlaz
    if m=0 then
            q=n; t=0; u=1;
    else
        NZDEXT(n mod m, m; q', t', u');
        q=q' ; t=t'; u=u'-t'⌊n/m⌋;
    end_if;
```

Složenost ista kao Euklidovog algoritma – $O(\log^3 M)$, $M = \max\{m,n\}$

Teorema: Za svaka dva prirodna broja m i n postoje brojevi t i $u \in \mathbb{Z}$ takvi da je $q = tn + u \cdot m$, $q = NZD(m,n)$.

– Ako su m i n uzajamno prosti tada je $tn + um = 1$

Posljedica: Neka su m i $n \in \mathbb{N}$ i $q = NZD(m,n)$. Tada m ima multiplikativno inverzni po modulu n akko je $q = 1$.

dokaz:

($\Leftarrow$): $q = 1$

$NZD(m,n) = 1 \xrightarrow{\text{proširenji Euk. alg.}} \exists t,u : tn + um = q = 1 \Rightarrow um = 1 - tn \mid \bmod n$

$\Rightarrow um \equiv 1 - tn \equiv 1 \pmod n$

$\Rightarrow u$ je inverzni za m tj. $u = m^{-1} \pmod n$

– i inverzni računamo za polinomno vrijeme.

($\Rightarrow$): m ima inverzni po modulu n

Pretpostavimo da je $q > 1$.

$\exists d$ tako da $d \cdot m \equiv 1 \pmod n$

$\exists c$ tako da $d \cdot m = cn + 1$

$\Rightarrow dm - cn = 1$

$q \mid m \wedge q \mid n \Rightarrow q \mid dm - cn \Rightarrow q \mid 1$ što je suprotno pretp.-

ostaci: $q > 1 \Rightarrow q = 1$. ///

! $\exists$ inverzni po modulu $n \Leftrightarrow NZD(m,n) = 1$

Prsten cijelih brojeva po modulu n

Poistovjećujemo $\mathbb{Z}$ i n fiksiran broj.

Ljudi u vidu n možemo uvesti kongurenciju

$$a \equiv b \pmod n \Leftrightarrow n \mid (a-b)$$

→ ovo je relacija ekvivalencije ⇒ skup se razbija na klase

a: $[a]_n = \{ b \in \mathbb{Z} \mid b \equiv a \pmod n \}$

Pitanje: Koliko je različitih klasa za različite n-ove?

$0, 1, \dots, n-1$

$0 \leq b, a < n \wedge a \neq b \Rightarrow 0 < |a-b| < n \Rightarrow n \nmid (a-b)$

k-proizvoljno $k = q \cdot n + r$, $0 \leq r < n$

Skup svih ovih klasa označavamo sa:

$\mathbb{Z}_n = \{ [k]_n \mid k \in \mathbb{Z} \}$ i $card(\mathbb{Z}_n) = n$

Imamo skup, pa uvodimo i operacije $+, \cdot$ na sledeći način:

$[a] + [b] = [a+b]$ → ovo je dobro definisano tj. ne zavisi od predstavnika a

$[a] \cdot [b] = [a \cdot b]$ → -"-

Umjesto $[1]$ koristimo oznaku $\bar{1}$ ili 1.

-"- $[a]$ -"- a ili $\bar{a}$, $0 \leq a < n$.

Skup $\mathbb{Z}_n = \{0, 1, \dots, n-1\}$

$(\mathbb{Z}_n, +, \cdot)$ je prsten = prsten cijelih brojeva po modulu n

Pitanje: Kad je ovaj prsten polje?

Primjer: $(Z_6, +, \cdot)$ je prsten, ali nije polje

$2 \cdot 3 = 0 \Rightarrow$ i 2 i 3 djelioci nule.

U_n - skup svih elemenata iz Z_n koji imaju multiplik. inverzni

$U_n = \{ a \in Z_n \mid \exists a^{-1} \}$

$U_n \equiv Z_n \setminus \{0\} \Rightarrow$ svi imaju inverzne $\Rightarrow$ u ovom slučaju struktura je polje $\mapsto$ ovo važi ako je n-prost broj tj.

$|U_n| = |Z_n \setminus \{0\}| = n-1$ tj. $|U_n| = \phi(n)$

Ciklične grupe

$A = \{a_1, \_, a_n\}$ (svi elementi moraju biti iz strukture koja je grupa)

$A^* = \{ a_1^{p_1} \cdot a_2^{p_2} \cdot \ldots \cdot a_n^{p_n} \mid p_i \in N \cup \{0\}, i = \overline{1,n} \}$

$(A^*, \cdot)$ - grupa generisana skupom A.

$A = \{a\}$
$A^* = \{ a^p \mid p \in N \cup \{0\} \}$
$\Rightarrow (A^*, \cdot)$ - ciklička grupa = grupa generisana sa jednim elementom

a - primitivnim korijenom

Teorema (Ferma): Neka su $b, n \in N$ takvi da je $NZD(b,n) = 1$ tada je $b^{\phi(n)} \equiv 1 \pmod n$.

dokaz: Red elementa dijeli red grupe

$|G|$ - red grupe

$b \in G$, minimalno k tako da je $b^k = 1$ je red elementa b.

Svaki element ima svoj red ako je grupa konačna (tj. $|G|$ je konačna) $\Rightarrow \exists k$ t.d. $k \mid |G|$

Posmatrajmo grupu $U_n = \{$ el. koji su uzajamno prosti sa $n\}$

$(U_n, \cdot)$ je grupa

$|U_n| = \phi(n)$

$NZD(b,n)=1 \Longrightarrow b \in U_n$

U_n - konačno $\Longrightarrow \exists k : b^k \equiv 1 \pmod n$

$\not\Longrightarrow k \mid \phi(n) \Rightarrow \phi(n) = k \cdot s$

$\not\Longrightarrow b^{\phi(n)} \equiv (b^k)^s \equiv 1^s \equiv 1 \pmod n$

Posljedica: (Mala Fermaova)

Ako je n - prost broj i $b \not\equiv 0 \pmod n$ onda je $b^{n-1} \equiv 1 \pmod n$ $\quad (b^n \equiv b \pmod n)$

Dokaz:

n - prost $\Rightarrow \phi(n) = n-1 \Rightarrow b^{\phi(n)} \equiv b^{n-1} \pmod n$ ///

Pitanje: Kad je U_n ciklička?

Teorema: U_n je ciklička grupa akko je

1) $n=2$ ili 2) $n=4$ ili 3) $n=p^a$, p-neparan prost br. ili

4) $n=2p^a$, p-neparan prost broj

Primjer: U_8 nije ciklička jer je $8=2^3$.

Posljedica: Ako je U_n ciklička grupa (i n neparan) tada jednačina $x^2=1$ ima samo rješenje $x=\pm 1$.

Primjer: $U_8 : x^2=1 \Rightarrow x \neq \pm 1$

Teorema: (Kineska teorema o ostacima)

Ako su $m_i \in \mathbb{Z}$ za $i=\overline{1,r}$ po parovima međusobno prosti brojevi i $M = m_1 \cdot \ldots \cdot m_r$ onda preslikavanje koje svakom $x \in \mathbb{Z}$ $(0 \le x \le M-1)$ pridružuje r-torku $(b_1, \ldots, b_r)$, gdje je $b_i \equiv x \pmod{m_i}$ je bijekcija između skupa $\mathbb{Z}_n$ i $\mathbb{Z}_{m_1} \times \ldots \times \mathbb{Z}_{m_r}$

Druga formulacija:

Ako su $m_i \in \mathbb{Z}$ za $i=\overline{1,r}$ po parovima međusobno prosti brojevi i $M = m_1 \cdot \ldots \cdot m_r$ i $x \in \mathbb{Z}$ $(0 \le x \le M-1)$ tada sistem j-na $b_i \equiv x \pmod{m_i}$, $i=\overline{1,r}$ $(b_i \in \mathbb{Z}_{m_i})$ ima jedinstveno rješenje x u $\mathbb{Z}_n$.

Primjer: $x \equiv 2 \pmod 3$, $x \equiv 4 \pmod 7$ $\Big\}$ $x = ?$

$0 \le x \le 3 \cdot 7 - 1 = 20$

dokaz: $F(x) = (b_1, \dots, b_r)$, $b_i \equiv x \pmod{m_i}$

? "1-1": $F(x) = F(y) \Rightarrow x = y$

$x \neq y \Rightarrow F(x) \neq F(y)$

$F(x) = F(y) \Rightarrow (b_1, \dots, b_r) = (a_1, \dots, a_r)$

$\Rightarrow b_i = a_i$

$x \equiv b_i \pmod{m_i}$, $y \equiv a_i \pmod{m_i}$ $\Big|\Rightarrow y \equiv b_i \pmod{m_i} \Rightarrow x - y \equiv 0 \pmod{m_i}$

$\Rightarrow m_i \mid (x-y)$, $i = \overline{1,r}$

$\Rightarrow m_1 \cdot \dots \cdot m_r \mid (x-y) \Rightarrow M \mid (x-y)$

$\Rightarrow x \equiv y \pmod M$

$0 \le x, y \le M-1 \wedge M \mid (x-y)$

$\Downarrow$

$x - y = 0 \Longrightarrow x = y$

? "na": Za $\forall (b_1, \dots, b_r)$ treba da $\exists x$ t.d. $F(x) = (b_1, \dots, b_r)$

$x \equiv \sum_{i=1}^{r} c_i b_i \pmod M$

$x \equiv b_i \pmod{m_i}$

$c_i \equiv 0 \pmod{m_j}$, $i \neq j$ $\Big|\Rightarrow c_i = d_i \cdot \frac{M}{m_i}$

$c_i = 1 \pmod{m_i}$

Sad treba izabrati d_i tako da važi

$d_i \frac{M}{m_i} \equiv 1 \pmod{m_i} \iff d_i \equiv \left(\frac{M}{m_i}\right)^{-1} \pmod{m_i}$

Pitanje: Postoji li $\left(\frac{M}{m_i}\right)^{-1}$?

$\frac{M}{m_i}$ i m_i su uzajamno prosti $\Rightarrow \exists u, t$

$u \cdot m_i + t \cdot \frac{M}{m_i} = 1$

$d_i = t_i$ – ovo je naše rješenje i mi ga možemo konstruisat

$c_i = t_i \frac{M}{m_i}$

$$x \equiv \sum_{i=1}^{n} t_i \frac{M}{m_i} \cdot b_i \pmod{M} \quad ///$$

$M = 3 \cdot 7 = 21$

$m_1 = 3 \quad m_2 = 7$

$\frac{M}{m_1} = 7 \equiv 1 \pmod 3 \implies t_1 = 1$

$\frac{M}{m_2} = 3 \equiv 1 \pmod 7 \implies t_2 = 5$

$x \equiv 1 \cdot 7 \cdot 2 + 5 \cdot 3 \cdot 4 \pmod{21}$

$x \equiv 14 + 60 \pmod{21}$

$x \equiv 74 \pmod{21}$

$\implies x = 11$

11. čas:

Test primalnosti:

Neka je $n \in \mathbb{N}$. Provjeriti da li je n-prost ili složen?
Možemo ga definisati i preko jezika: $L=\{p \in \mathbb{N} / p\text{-prost}\}$. Za $n \in \mathbb{N}$ treba provjeriti da li je $n \in L$? Odgovor: Da ili Ne.
Pitanje: Kojoj klasi ovaj zadatak pripada?
- Jasno je da pripada NP! Ako je n-složen, mi možemo pogoditi dva broja a i b i provjeriti da li je $n=a \cdot b$ - polinomijalno.
Do skora se nije znalo da li je P ili NP-težak: Mislilo se da je u nekoj klasi između. Danas se zna da ovaj jezik pripada klasi P (2002.g.).

Pseudotestovi primalnosti:

Za date brojeve $b, n \in \mathbb{N}$ treba provjeriti da li je n-složen broj koristeći bazu b? (Broj b: $1<b<n$)
Odgovori su: n je složen i neodlučeno
Ako je odgovor n je složen onda je n zaista složen - daje tačne odgovore.
Ovaj algoritam za polinomijalno vrijeme daje odgovor.
Navešćemo nekoliko testova:
test 1°: Za dati ulaz b, n odgovoriti n je složen ako b dijeli n inače odgovori neodlučan.
Primjer: ulaz (3, 48) — odgovor n je složen jer 3/48
ulaz (5, 48) — odgovor neodlučan
Pitanje: Sa kojom vjerovatnoćom biramo b da odgovor bude složen je?
$n = p^2$, p-prost.
Koliko brojeva dijeli ovo n, a da su >0, $a < p^2$?

Odgovor: samo jedan broj p : $1<p<n$ i p/n

Različitih baza ima p^2-2.

Vjerovatnoća ~~ima~~ je $\frac{1}{p^2-2}$. To je mala vjerovatnoća — ovaj test zato nije dobar.

test 2°: Za dati ulaz (b,n) odgovoriti n je složen ako je $NZD(b,n)>1$, inače odgovoriti neodlučan.

Kolika je vjerovatnoća da izaberemo b, da odgovor bude n je složen?

$n=p^2$ $\quad 1\cdot p, 2p, \dots, (p-1)p \Rightarrow$ Povoljnih brojeva je $p-1$.

p^2-2 - ukupan broj kandidata

$\frac{p-1}{p^2-2}$ - vjerovatnoća

$\frac{p-1}{p^2-2} \sim \frac{p-1}{p^2-1} = \frac{1}{p+1} \Rightarrow$ bolji od prethodnog, ali nije dobar

test 3°: Za dati ulaz (b,n) odgovoriti n je složen ako je $NZD(b,n)>1$ ili $b^{n-1} \not\equiv 1 \pmod n$, inače odgovori neodlučno.

vjerovatnoća je $< \frac{1}{2}$.

test 4°: (Strogi test)

Neka je dat ulaz (b,n) takav da je $n-1=2^q \cdot m$, gdje je m - neparan prirodan broj, $q \neq 1$; ako je zadovoljen jedan od sledeća dva uslova:

1) $b^m \equiv 1 \pmod n$

2) $(\exists i \in \{0,\dots,q-1\})$ t.d. je $b^{m\cdot 2^i} \equiv -1 \pmod n$

tada odgovori neodlučan je, inače odgovori n je složen.

dokaz:

Pretpostavimo suprotno tj. n-prost, a test odgovara n je složen (nisu ispunjeni ni 1) ni 2))

$\xrightarrow[\text{teor.}]{\text{Mala Ferm.}} b^{n-1} \equiv 1 \pmod n$

Teor.: n-neparan prost onda j-na $x^2 = 1 \pmod n$ ima samo r-nja

$x = \pm 1$

$\Rightarrow b^{2^{q-1}\cdot m} \equiv 1 \pmod n \Rightarrow (b^{2^{q-1}\cdot m})^2 \equiv 1 \pmod n$ - ova j-na ima dva rješenja tj.

$\underbrace{b^{2^{q-1}\cdot m} \equiv -1 \pmod n}$ v $b^{2^{q-1}\cdot m} \equiv 1 \pmod n$

ovo neće biti jer bi onda bilo ispunjeno 2) što nije po pretpostavci tačno

$b^{2^{q-1}\cdot m} \equiv 1 \pmod n \xrightarrow[\text{indukcijom}]{\text{inverznom}} b^{2^k \cdot m} \equiv 1 \pmod n$, $1 \le k < q \Rightarrow$

$b^{2^{k-1}\cdot m} \equiv \pm 1 \pmod n \rightarrow$ otpada slučaj (-1) zbog istog razloga kao prethodno

$\Rightarrow b^{2m} \equiv 1 \pmod n \Leftrightarrow (b^m)^2 \equiv 1 \pmod n$ - dva rešenja tj.

$b^m \equiv \pm 1 \pmod n$

$\Rightarrow$ $b^m \equiv -1 \pmod n$ - otpada zbog 2)

$b^m = 1 \pmod n$ - otpada zbog 1.) | kontradikcija tj. pretpostavka nije dobra

Teorema: Neka je B'-skup cijelih brojeva iz intervala $[1, n-1]$ takvih da algoritam za ulaz (b, n) odgovori neodlučeno $(b \in B')$. Ako je n-složen tada B' sadrži najviše polovinu cijelih brojeva iz intervala $[1, n-1]$.

$B' = \{b \in [1, n-1] \mid$ za (b,n) odgovor je neodlučeno$\}$

n-složen $\Rightarrow |B'| \le \frac{n-1}{2}$

Sa kojom vjerovatnoćom možemo izabrati b da za n-složen odgovor bude n je složen?

vjerovatnoća je $\ge \frac{1}{2}$.

100 puta slučajno birati bazu b i primjeniti test 4° za par (b,n). Pitanje ako za bar jedno b dobijemo n je složen, prekinemo

i odgovore je složen, a u suprotnom sa vjerovatnoćom $1 - \frac{1}{2^{100}}$ možemo tvrditi da je n-prost (velika vjerovatnoća)

RSA-algoritam

Koristi se za šifrovanje javnim ključem. Ovaj algoritam se bazira na "teškoj" faktorizaciji velikih brojeva.

Opšta ideja: Imamo pošiljaoca i primaoca; pošiljaoc šalje poruku, a primaoc prima; i prisluškivača koji čita tu poruku.

Pošiljaoc i primaoc se dogovore o tajnom ključu.

ALGORITAM → QOALTARIM → ALGORITAM
(poruka pošiljaoca)

kodiranje – dekodiranje

Prvi algoritam sa javnim ključem je RSA-algoritam. Naziv je dobio po imenima svojih tvoraca.

Primalac daje javni ključ i njega svi znaju, svako može da mu šalje poruku, ali neće moći saznati šta su oni međusobno poslali tj. oni svi mogu šifrovati, ali neće moći dešifrovati.

Primaoc generiše i tajni ključ.

Algoritam za generisanje ključeva je:

1° korak: Generišemo dva velika prosta broja p i q, koji približno imaju jednak broj bita i zahtijeva se da $n = p \cdot q$ ima unaprijed zadat broj bita tj. broj bita za N je b tj. $|n| = b$ onda p i q imaju po $b/2$ bita.

Slučajno generišemo bite P_1, prva dva 1, ostale slučajno i posljednji 1 (zbog neparnosti) i da ima $b/2$ bita.

Provjeravamo da li je prost:

1° jeste – ok!

2° ako nije uvećamo ga za 2 i sve tako dok

ne postane prost. (dovoljno je da bude i pseudoprost)
Kad na ovaj način završimo generisanje P, pređemo na q koji generišemo na isti način.

2° korak: Izračunati: $n = p \cdot q$, $\phi(n) = (p-1)(q-1)$ - Ojlerova funkcija (možemo jer su p i q prosti)

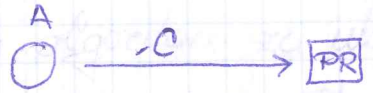
3° korak: Izabrati $e \in N$, pri čemu je $1 < e < \phi(n)$ tako da je $NZD(e, \phi(n)) = 1$ i e je obično oblika $e = 2^{2^k} + 1$ (ovaj oblik nam omogućava brže stepenovanje)

4° korak: Izračunati tajni eksponent $d \in N$, $1 < d < \phi(n)$ (ne preporučuje se 1) takav da je $e \cdot d \equiv 1 \pmod{\phi(n)}$ ($\Leftrightarrow$ d je inverz od e po modulu $\phi(n)$) $\Rightarrow d \equiv e^{-1} \pmod{\phi(n)}$

$\Longleftrightarrow$ ovim smo završili generisanje ključeva

5° korak: Kao javni ključ objavimo par (n, e), a kao tajni ključ (ključ za dešifrovanje) je (n, d). U tajnosti čuvamo i vrijednosti $p, q, \phi(n)$ (da nam neko ne izračuna d)
Za sve imamo polinomijalno vrijeme.

Pitanje: Kako nam neko (A) šalje poruku ako zna (n, e)?



A svoju poruku kodira brojevima: A 11, B 12, ..., Ž 40

primjer: ALGORITAM — 1122...11...

Ako je broj veći mi ga razbijemo na brojeve koji imaju bita kao i n, ali koji su manji od n i kodiramo uzastopno takve poruke.

Posmatrajmo jedno m koje je naša poruka t.d $m < n$ i $|m|$
Pošiljaoc A koji ima poruku m računa $c \equiv m^e \pmod{n}$ i šalje onda poruku c. (KORAK šifrovanja)

Korak dešifrovanja:

dobija poruku c i racuna $x \equiv c^d \pmod n$.

Ispostavlja se da je $x = m$ odnosno x je orginalna poruka

dokaz: Dokazujemo da je $m = x$

$de \equiv 1 \pmod{\phi(n)} \iff de = k \cdot \phi(n) + 1$

b-proiz. $NZD(b,n) = 1 \Rightarrow b^{\phi(n)} \equiv 1 \pmod n$ (Mala Ferm. teor.)

$\Rightarrow x \equiv c^d \equiv (m^e)^d \equiv m^{ed} \equiv m^{k\phi(n)+1} \equiv (m^{\phi(n)})^k \cdot m \equiv$

[m i n uzajamno prosti (velika vjerovatnoca jer su veliki brojevi)]

$\equiv 1^k \cdot m \equiv m \pmod n$

///

Stepenovanje je polinomijalno $\Rightarrow$ šifrovanje i desifrovanje je polinomijalno.

Prisluškivač zna (n,e), a treba da zna $p, q, \phi(n)$ – to znači mora da zna faktorizaciju $n = p \cdot q$ – a za to nema polinomijalnog algoritma, pa on nemože da čita našu poruku za kratko vrijeme — ono je za njega tajno.

$F_{14} = 2^{2^{14}} + 1$ – složen, ali faktori mu se ne znaju.

Problem RSA-alg. je nepokretna tačka (poruka se slika sama u sebe, što nije pogodno), ali postoje načini da se to prevaziđe.