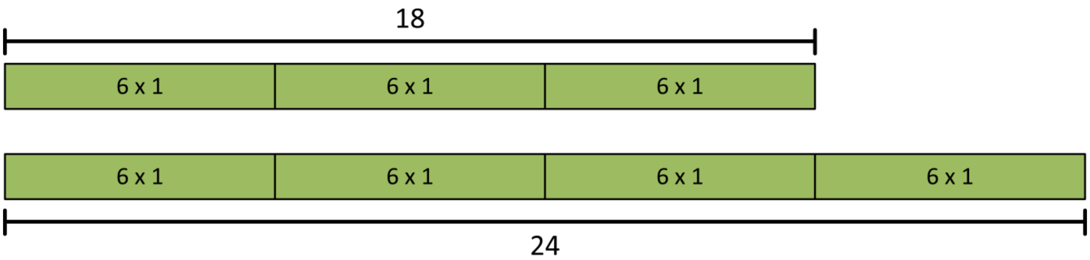
Euklidov algoritam

Sada ćemo naučiti šta je to najveći zajednički delilac/najmanji zajednički sadržalac i kako se oni izračunavaju. Za početak, posmatrajmo sledeći problem

Problem 1 . Date su dve table (trake) dimenzija $a \times 1$ i $b \times 1$. Odrediti najveći broj d tako da je obe table moguće potpuno "popločati" (bez preklapanja) dominama dimenzija $d \times 1$.

Primetimo da je dominama 1×1 uvek moguće popločati date table tj. broj d sa traženim osobinama sigurno postoji ($d = 1$) ali kako naći najveći od njih? Za početak, možemo analizirati manje table i "osetiti" kako broj d treba da izgleda u funkciji od a i b.



Slika 1: Posle malo eksperimentisanja, može se pokazati da je za $a = 18$ i $b = 24$ rešenje $d = 6$, tj. dominama 6×1 je moguće popločati i tablu 18×1 i tablu 24×1 dok većim dominama to nije moguće.

Očigledno, mora važiti $d|a$ i $d|b$. Sa druge strane ovo je i dovoljno da bi popločavanje bilo moguće. Prema tome, rešenje je najveći broj koji deli i a i b. Taj broj ima značajnu ulogu u mnogim problemima teorije brojeva i nosi poseban naziv:

Definicija 1 (NZD). Za date cele brojeve a i b, koji nisu oba istovremeno jednaka nuli, najveći ceo broj d za koji važi $d|a$ i $d|b$ se naziva **najveći zajednički delilac** brojeva a i b i označava se sa $NZD(a, b)$ ili $\gcd(a, b)$ ili jednostavno (a, b).

Nadalje ćemo umesto "najveći zajednički delilac" pisati skraćeno NZD. Kao primer, imamo da važi $(24,18) = 6$, $(-22, 33) = 11$, $(63, 100) = 1$, $(2013, 0) = 2013$. Takođe, ukoliko važi $(a, b) = 1$, tada za brojeve a i b kažemo da su **uzajamno prosti**, iako oni sami ne moraju biti prosti (o čemu svedoči primer $a = 63, b = 100$). Ukoliko sa $D(x)$ označimo skup delioca broja x tada je (a, b) najveći element skupa $D(a) \cap D(b)$. Ranije smo pomenuli da u problemima deljivosti znak broja ne igra neku posebnu ulogu – u ovom slučaju važi $(a, b) = (|a|, |b|)$ pa ćemo uglavnom posmatrati nenegativne cele brojeve. Ako su a, b prirodni brojevi, nije teško primetiti da važi

1. $(a, b) = (b, a)$,
2. $1 \leq (a, b) \leq \min(a, b)$,
3. $(a, 0) = (a, a) = a$.

Osobina broj 2 je posebno interesantna: ona je tačna jer $1 \in D(a) \cap D(b)$ (leva nejednakost) i činjenice da su delioci prirodnih brojeva ne veći od njih samih (desna nejednakost). Ova osobina nam govori da je za čuvanje (a, b) dovoljan isti tip podataka koji koristimo za a i b. Jedan vrlo jednostavan algoritam za računanje NZD-a dva broja je sledeći: prođimo kroz sve (pozitivne) delioce manjeg broja i vratimo najveći od njih koji deli veći broj. Na osnovu prethodnih delova lekcije, ovakvo računanje

(a, b) je složenosti $O(\sqrt{\min(a, b)})$. Međutim, za računanje NZD-a postoji mnogo brži (i jednostavniji!) algoritam. Poznat je pod nazivom **Euklidov algoritam** i baziran je na sledećoj vrlo jednostavnoj teoremi:

Teorema 1 . Ukoliko za cele brojeve a, b, q i r važi $a = bq + r$, tada je $(a, b) = (b, r)$.

Dokažimo ovu teoremu. Ukoliko je d proizvoljan zajednički delilac brojeva a i b, tj. $d|a$ i $d|b$, tada $d|a - bq = r$ pa je d ujedno i zajednički delilac brojeva b i r. Obratno, ako je d zajednički delilac brojeva b i r, tada $d|bq + r = a$ pa je d zajednički delilac brojeva a i b. Prema tome, skupovi zajedničkih delilaca brojeva a i b i brojeva b i r su jednaki pa su im jednaki i najveći elementi tj. $(a, b) = (b, r)$.

Kako nam prethodna teorema pomaže? Ako su $a \geq b$ prirodni brojevi i r_1 ostatak pri deljenju broja a brojem b, tada je (prema teoremi o deljenju sa ostatkom) $a = bq + r_1$ za neko $q \in N$. Prema Teoremi 1 važi $(a, b) = (b, r_1)$. Takođe, važi i $r_1 < b$ (jer $r_1 \in [0, b - 1]$), tj. sada smo problem sveli na nalaženje NZD-a za manje brojeve. Ukoliko je $r_1 \neq 0$ i r_2 ostatak pri deljenju broja b brojem r_1, tada, primenjujući istu teoremu, dobijamo $(b, r_1) = (r_1, r_2)$. Ukoliko je $r_2 \neq 0$ i r_3 ostatak pri deljenju r_1 sa r_2 i analogno za r_4, r_5, ... pri čemu je r_n prvi ostatak koji je jednak 0, tada je

$$(a, b) = (b, r_1) = (r_1, r_2) = (r_2, r_3) = (r_3, r_4) = \cdots = (r_{n-1}, r_n) = r_{n-1}.$$

Zaista, kako se radi o ostacima, važi $b > r_1 > r_2 > r_3 > \cdots$ pa će se ovaj strogo opadjući niz prirodnih brojeve završiti tj. pojaviće se nula. Praktično, u svakom koraku koristimo jednakost

$$(a, b) = (b, a \bmod b)$$

i kada druga koordinata uređenog para postane nula, prva koordinata je traženi NZD. Demonstrirajmo ovo na jednostavnom primeru:

$$(30, 21) = (21, 30 \bmod 21) = (21, 9) = (9, 21 \bmod 9) = (9, 3) = (3, 9 \bmod 3) = (3, 0) = 3.$$

Ovo je cela filozofija Euklidovog algoritma koji se uglavnom implementira u iterativnoj ili rekurzivnoj varijanti. Sledeći pseudokod je iterativna varijanta:

```
=================================================================
01     function NZD(int a, int b) : int  // Euklid (iterativni)

02             while (b ≠ 0) do
03                     tmp ← a;
04                     a ← b;
05                     b ← tmp mod b;
06             end while

07             return a;

08     end function
=================================================================
```

Linije 03-05 predstavljaju zamenu $(a, b) \rightarrow (b, a \bmod b)$. U rekurzivnoj varijanti (pogledati lekciju **Rekurzija**) algoritam bi izgledao ovako:

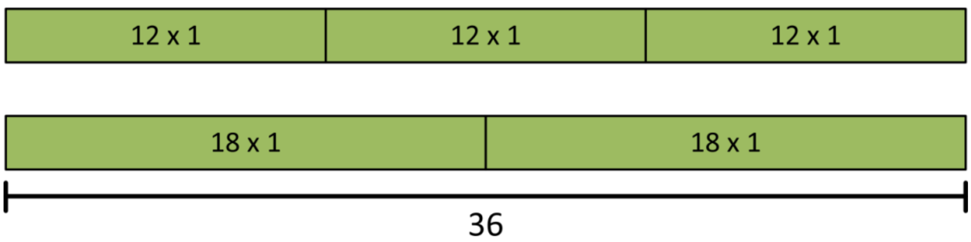
```
=================================================================
function NZD(int a, int b) : int  // Euklid (rekruzivni)

      if (b = 0) then
            return a;
      else
            return NZD(b, a mod b);
      end if

end function
=================================================================
```

Na primeru $a = 30$, $b = 21$, imali bismo rekurzivne pozive $(30{,}21) \rightarrow (21{,}9) \rightarrow (9{,}3) \rightarrow (3{,}0)$. Zbog korišćenja operacije mod, funkciju NZD (i iterativnu i rekurzivnu varijantu) je potrebno pozivati **isključivo sa nenegativnim brojevima**. Ukoliko radimo sa celim brojevima a i b, najjednostavnij rešenje je pozivati funkciju kao $NZD(|a|,|b|)$. Složenost Eklidovog algortima je $O(\log \min(a,b))$ pa se može primeniti na jako velike brojeve (npr. prethondi algoritam složenosti $O\left(\sqrt{\min(a,b)}\right)$ ne može raditi sa brojevima reda veličine 10^{18} dok za Euklida to nije problem). Analiza složenosti Euklidovog algoritma nije jednostavna – videti Napomenu 1.

Sada kada znamo efikasan algoritam za izračunavanje NZD-a dva broja, znamo i da rešimo Problem 1. Zato pređimo na "obrnuti" problem:

Problem 2. Odrediti najmanji broj s tako da je tablu (traku) dimenizije $s \times 1$ moguće potpuno popločati (bez preklapanja) i (samo) dominama dimenzije $a \times 1$ i (samo) dominama dimenzije $b \times 1$.

Kao i u analizi prethodnog problema, nije teško zaključiti da nam treba najmanji prirodan broj s tako da $a|s$ i $b|s$. Primetimo da za $s = ab$ važi $a|s$ i $b|s$, što znači da rešenje uvek postoji.



Slika 2: Za $a = 12$ i $b = 18$ uz malo eksperimentisanja dolazimo do $s = 36$, tj. tabla dimenzije 36×1 se može popločati koriteći samo domine 12×1i koristeći samo domine 18×1 i ne postoji manja tabla sa ovom osobinom.

Definicija 2 (NZS). Za date cele brojeve a i b, različite od nule, najmanji ceo broj s za koji važi $a|s$ i $b|s$ se naziva **najmanji zajednički sadržalac** brojeva a i b i označava se sa $NZS(a,b)$ ili $lcm(a,b)$ ili jednostavno $[a,b]$.

Dakle NZS je, na neki način, "inverzan" NZD-u. Kao primere, imamo $[12,18] = 36$, $[-22,33] = 66$, $[21,10] = 210$, $[2013{,}1] = 2013$. Kao i kod NZD-a, i ovde ćemo koristiti skraćenicu NZS i koncentrisaćemo se na pozitivne brojeve. Neke od poznatijih osobina NZS-a su $(a, b \in N)$:

1. $[a,b] = [b,a]$,
2. $\max(a,b) \leq [a,b] \leq \mathrm{ab}$,
3. $[a,1] = [a,a] = a$.

Ne samo da je NZS veći i od a i od b (jer ih sadrži) već može biti reda veličine ab. Ovo ima za posledicu sledeća dva problema: 1) za velike vrednosti a i b, $[a, b]$ može ispasti iz opsega korišćenog tipa podataka (*overflow*) i 2) ne možemo jednostavno iterirati po svim brojevima manjim od ab da bismo pronašli $[a, b]$ jer je to previše sporo. Srećom, NZD ne pati od ovakvih problema i sledeća teorema daje jako lepu vezu između (a, b) i $[a, b]$:

Teorema 2 . Za cele brojeve a i b, različite od nule, važi $(a, b) \cdot [a, b] = |ab|$.

Za dokaz, videti Napomenu 2. Ova lepa formula nam daje par korisnih informacija: za proizvoljne $a, b \in N$, važi $(a, b) | ab$ (očigledno), $[a, b] | ab$ (manje očigledno) i a i b su uzajamno prosti ako i samo ako je $[a, b] = ab$. Naravno, nama je od najvećeg značaja činjenica da je $[a, b] = \frac{ab}{(a,b)}$, tj. NZS se direktno računa na osnovu NZD-a:

```
=================================================================
01     function NZS(int a, int b) : int
02            return (a / NZD(a,b)) * b;
03     end function
=================================================================
```

Složenost je, naravno, ista kao i složenost Euklidovog algoritma - $O(\log \min(a, b))$. Iako je glavni deo koda stao u svega jednu liniju, u toj liniji postoji jedan suptilan deo: primetimo da NZS računamo kao $\frac{a}{(a,b)} \cdot b$ a ne kao $\frac{ab}{(a,b)}$ iako su ova dva načina ekvivalentna. Razlog je da bi međurezultat stalno bio manji ili jednak $[a, b]$. Npr. za $a = b = 10^{18}$ je $[a, b] = 10^{18}$ i svi brojevi staju u 64-bitni tip podataka. Ukoliko NZS računamo kao $\frac{ab}{(a,b)}$ imaćemo međurezultat $10^{18} \cdot 10^{18}$ koji će ispasti iz 64-bitnog tipa i doći će do greške; sa druge strane, redosled $\frac{a}{(a,b)} \cdot b$ obezbeđuje korektan rezultat.

Definicije NZD-a i NZS-a se prirodno mogu proširiti i na više od dva broja:

Definicija 3 . Najveći zajednički delilac celih brojeva $a_1, a_2, \dots, a_n$, u oznaci $(a_1, a_2, \dots, a_n)$, je najveći ceo broj koji deli svaki od njih.

Definicija 4 . Najveći zajednički sadržalac celih brojeva $a_1, a_2, \dots, a_n$, različitih od nule, u oznaci $[a_1, a_2, \dots, a_n]$, je najmanji prirodan broj koji je deljiv svakim od njih.

Npr. $(12, 8, 18) = 2$ i $[12, 8, 18] = 72$. Vidimo da su definicije NZD-a i NZS-a za dva broja samo specijalni slučajevi definicije za više brojeva. Već pomenute osobine se mogu generalizovati:

$$1 \le (a_1, a_2, \dots, a_n) \le \min(a_1, a_2, \dots, a_n),$$

$$\max(a_1, a_2, \dots, a_n) \le [a_1, a_2, \dots, a_n] \le a_1 \cdot a_2 \cdot \dots \cdot a_n.$$

Primetimo da NZS može biti jako veliki broj. Napomenimo da je u opštem slučaju, za $n > 2$, $(a_1, a_2, \dots, a_n) \cdot [a_1, a_2, \dots, a_n] \ne |a_1 \cdot a_2 \cdot \dots \cdot a_n|$. Srećom, važe sledeće rekurzivne formule:

$$(a_1, a_2, \dots, a_n) = \big((a_1, a_2), a_3, \dots, a_n\big),$$

$$[a_1, a_2, \dots, a_n] = \big[[a_1, a_2], a_3, \dots, a_n\big].$$

Prva formula nam govori da NZD n brojeva možemo izračunati tako što prvo izračunamo NZD neka dva od njih, zatim NZD dobijenog rezltata i nekog od preostalih brojeva, zatim NZD dobijenog rezultata i nekog od preostalih itd. Preciznije, prvo izračunamo $x_1 = (a_1, a_2)$, zatim $x_2 = (x_1, a_3)$, zatim $x_3 = (x_2, a_4)$ itd. Konačna vrednost je $(a_1, a_2, \ldots, a_n) = x_{n-1} = (x_{n-2}, a_n)$. Na prethodnom primeru to bi izgledalo $(12, 8, 18) = \big((12, 8), 18\big) = (4, 18) = 2$.

```
=================================================================
01     function NZD_Multi(int a[], int n) : int  //

02            x ← a[1];
03            for i ← 2 to n do
04                   x ← NZD(x, a[i]);
05            end for
06            return x;

07     end function
=================================================================
```

Na osnovu druge formule, potpuno ista priča važi za NZS s tim što u liniji 04 prethodnog pseudokoda treba staviti “$x \leftarrow NZS(x, a[i])$;”. Složenost izračunavanja NZD/NZS za n brojeva je $O(n \cdot \log MaxA)$ gde je $MaxA$ najveći od datih n brojeva (često je “pravi broj izvršenih operacija” mnogo manji). Zbog činjenice da NZS može biti jako veliki broj (naročito ako tražimo NZS za više od dva broja), u nekim zadacima se garantuje da su test primeri takvi da NZS staje u odgovarajući tip podataka iako na osnovu samih ograničenja to ne možemo zaključiti.

Napomena 1. Dokaz složenosti Euklidovog algoritma nije trivijalan i koristi Fibonačijeve brojeve. Preciznije, važi: Ako je $a > b \geq 1$ i $b < F_{k+1}$ (gde je F_n n-ti Fibonačijev broj) tada funkcija $NZD(a, b)$ napravi manje od k rekurzivnih poziva. Uzastopni Fibonačijevi brojevi su “najteži” za računanje, tj. za računanje $NZD(F_{k+1}, F_k)$ se napravi tačno $k - 1$ rekurzivnih poziva.

Napomena 2. Za prirodne brojeve a i b, neka je $\{p_1, p_2, \ldots, p_k\}$ skup njihovih zajedničkih prostih delilaca. Tada brojeve a i b možemo zapisati kao

$$a = p_1^{a_1} p_2^{a_2} \cdots p_k^{a_k}, \qquad b = p_1^{b_1} p_2^{b_2} \cdots p_k^{b_k},$$

pri čemu neki od a_i ili b_i mogu biti jednaki 0 (ukoliko p_i ne deli odgoravajući broj). **Jako je korisno razumeti** (a lako za dokazati) da tada važe sledeće jednakosti

$$(a, b) = p_1^{\min(a_1, b_1)} p_2^{\min(a_2, b_2)} \cdots p_k^{\min(a_k, b_k)},$$

$$[a, b] = p_1^{\max(a_1, b_1)} p_2^{\max(a_2, b_2)} \cdots p_k^{\max(a_k, b_k)}.$$

Na osnovu ovih jednakosti direktno sledi $(a, b) \cdot [a, b] = |ab|$.