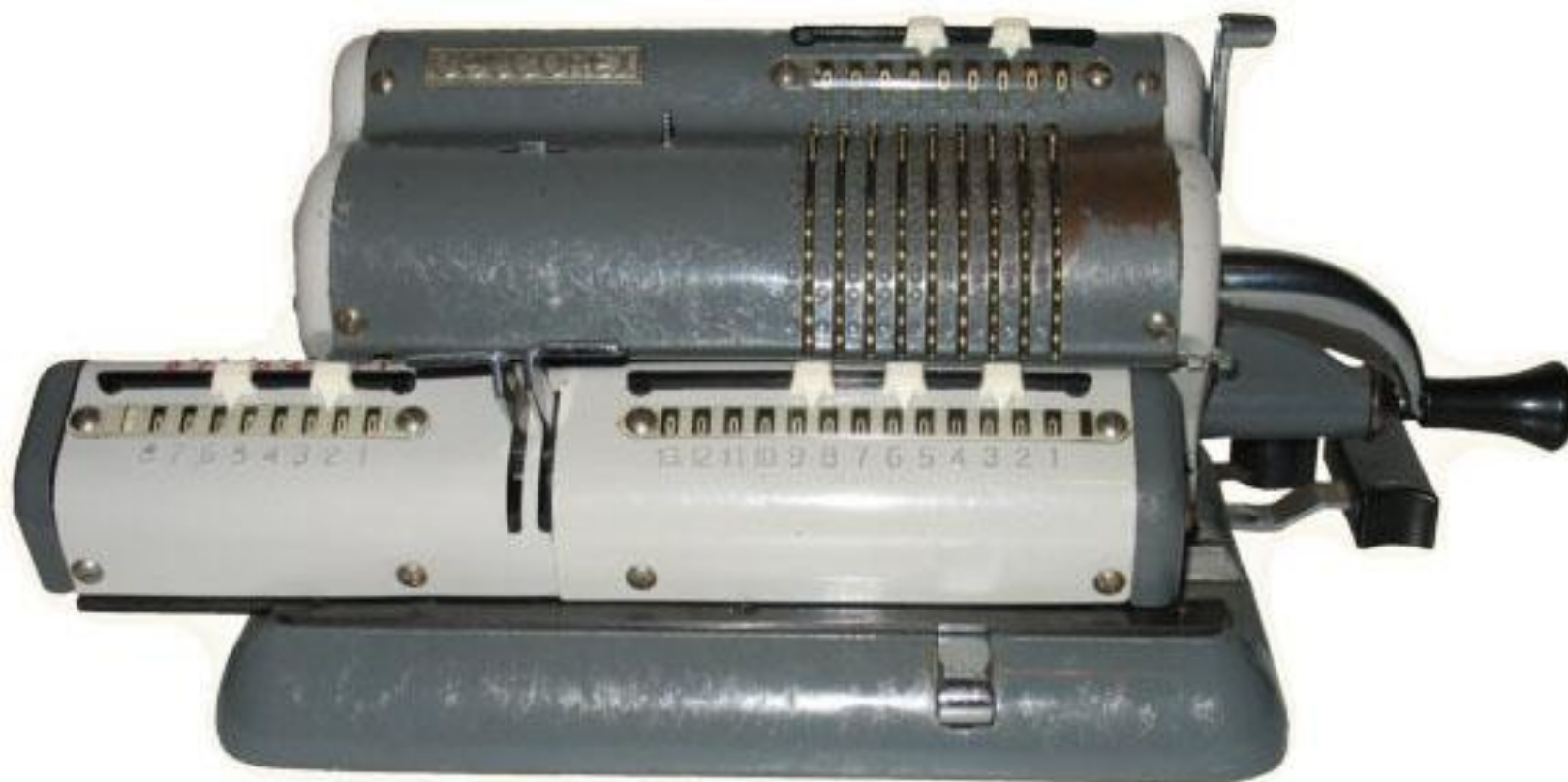


Algoritmi i strukture podataka

- predavanja -

1. Uvod



■ ZBRAJANJE: $123 + 456$

1. Resetiramo računalu (6)(7)

2. Unesemo 123 (1)

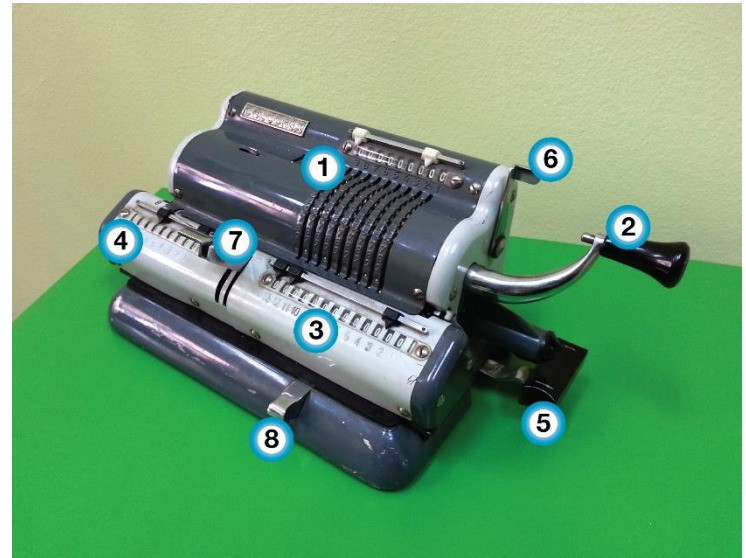
3. Okrenemo ručicu (2) u smjeru kazaljke na satu

■ u akumulatoru (3) je 123

1. Unesemo 456 (1)

2. Okrenemo ručicu (2) u smjeru kazaljke na satu

■ u akumulatoru(3) je 579



■ ODUZIMANJE: 456-321

1. Resetiramo računalu (6)(7)

2. Unesemo 456 (1)

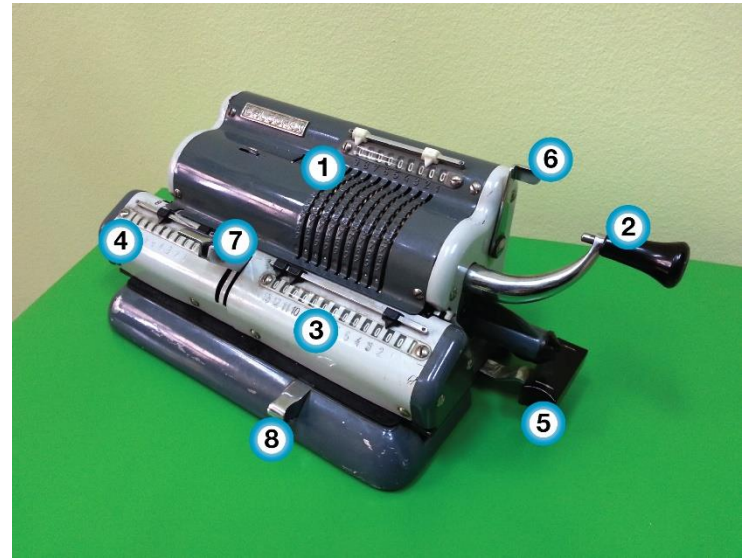
3. Okrenemo ručicu (2) u smjeru kazaljke na satu

■ u akumulatoru (3) je 456

1. Unesemo 321 (1)

2. Okrenemo ručicu (2) u smjeru **suprotnom** od kazaljke na satu

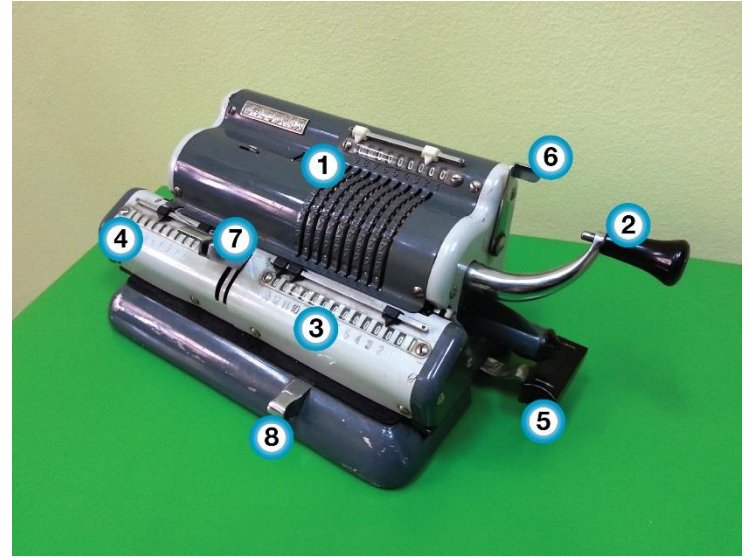
■ u akumulatoru (3) je 135



■ MNOŽENJE: $35 * 28$

1. Resetiramo računalu (6)(7)
2. Unesemo 35 (1)
3. Okrećemo ručicu (2) u smjeru kazaljke na satu dok brojač (4) ne pokazuje brojku 28.

■ u akumulatoru je 980



- **DIJELJENJE 22: 7 = 3.14285714 i ostatak 2**

1. Resetiramo računalu (6)(7)

2. Unesemo 22 (1)

3. Okrenemo ručicu (2) u smjeru kazaljke na satu

- **u akumulatoru (3) je 22**

1. Unesemo 7 (1)

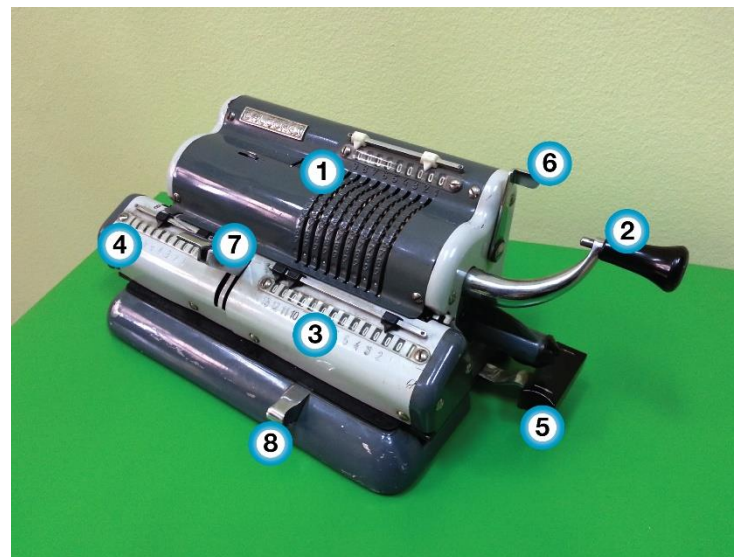
2. Resetiramo brojač (**SAMO LIJEVA RUČICA 7!**)

3. Okrećemo ručicu (2) u smjeru **suprotnom** od kazaljke na satu dok u akumulatoru ne ostane broj manji od djelitelja

4. pomaknemo akumulator za 1 mjesto lijevo sa ručicom (5)

5. Ponavljamo korake (6) i (7) dok ne izračunamo zadovoljavajući broj decimala

- **brojač (4) prikazuje 3,14285... u akumulatoru(3) se nalazi "ostatak"**



Zbroj prvih N prirodnih brojeva

- $1 + 2 + 3 + 4 + 5 + 6 + \dots + N-1 + N$

- For petlja?

```
#include <iostream>
using namespace std;

int main() {
    int n = 20;
    int suma = 0;
    for (int i = 1; i<=n; i++)
        suma += i;
    cout << suma;
}
```

- Gaussova formula: $N * (N+1) / 2$

Nalazi li se neki broj u zadanom polju brojeva?

- Ako polje **NIJE** sortirano
 - for petlja
 - Koliko će u prosijeku trajati traženje broja?
- Što ako polje **JE** sortirano?
- [1,2,3,5,8,23,1346,1347,1348,1450,23821,23825,23871...]
 - for petlja
 - Može li bolje?

Općenito o predmetu

- Osnovni koncepti struktura podataka i algoritama nad njima
- Koristit će se programski jezik C++
- Pretpostavke:
 - operativno znanje programiranja u C-u
 - poznavanje osnovnih koncepata OOP-a

Općenito o predmetu

- Okvirni sadržaj:
 - Uvod u C++ i dinamička rezervacija memorije
 - Složenost algoritama
 - Rekurzija
 - Sortiranje
 - Dinamičke strukture podataka (stog, red, lista, stablo, skup, mapa, gomila)
 - Raspršeno adresiranje (*hashing*)
 - Grafovi
 - Nizovi i obrada nizova

Tehnički detalji o predmetu

- Na web stranicama predmeta:
https://www.fer.unizg.hr/predmet/asp_b
- u mapi Materijali/Upute

Algoritam

- nedvosmisleno opisan način rješavanja nekog problema
- Algoritam mora:
 - Imati **početne objekte** nad kojima se obavljaju operacije
 - Rezultirati **završnim objektima** (rezultatima)
 - Biti **precizan**: svaki korak mora biti jasno opisan
 - Biti **konačan**: mora završiti nakon konačnog broja koraka/instrukcija
 - Biti **ispravan**: za svaki početni objekt mora dati točan rezultat
 - Biti **djelotvoran**: mora dati rezultat u konačnom vremenu, čak i kad bi se izvodio koristeći olovku i papir

Algoritam

- **precizno** opisan način rješavanja nekog problema
- Svojstva:
- **jednoznačno** određuje što treba napraviti
- moraju biti definirani **početni objekti** koji pripadaju nekom razredu objekata na kojima se obavljaju operacije
 - kao ishod algoritma dobiju se **završni objekti** ili **rezultati**
 - konačni broj koraka, gdje je svaki korak opisan **instrukcijom**
 - obavljanje je **algoritamski proces**
- **ispravan**, ako za svaki ulazni podatak, računa točan rezultat (izlazni podatak)
- **djelotvoran**, ako se dobije rezultat u konačnom vremenu i korištenjem konačnog memorijskog prostora

Svojstva algoritma - djelotvornost

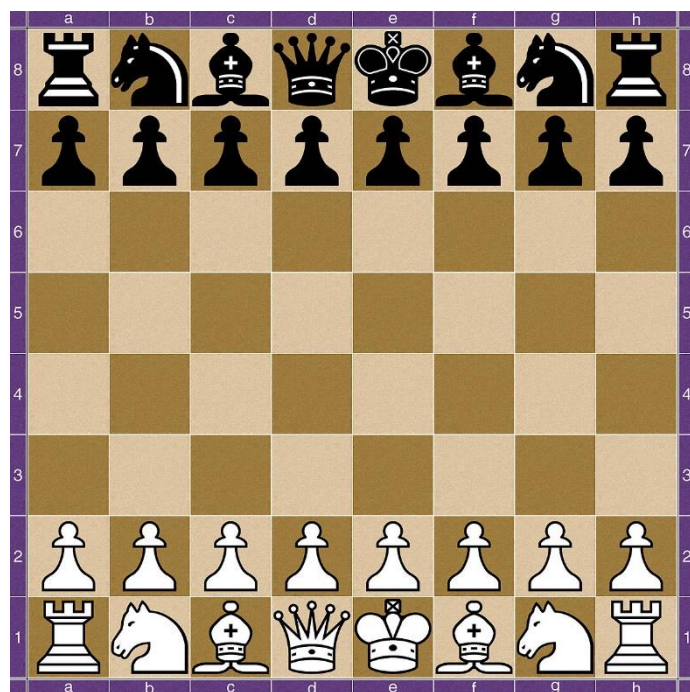
- algoritam mora biti **djelotvoran** (engl. *effective*)
 - u konačnom vremenu može se dobiti rezultat koristeći olovku i papir
- primjeri
 - zbrajanje cijelih brojeva je djelotvorno
 - zbrajanje realnih brojeva nije djelotvorno, jer se može pojaviti broj s beskonačno mnogo znamenki
- sa znanjem programiranja i uz razumijevanje problema koji rješava, student može napisati **djelotvoran** algoritam
→ je li to dovoljno?

Svojstva algoritma - učinkovitost

- cilj ovog predmeta je naučiti kako se oblikuje i programira **učinkovit** (engl. *efficient*) algoritam
 - učinkovit s obzirom na utrošene resurse - vrijeme, procesor, disk, memoriju
 - npr. vrijeme izvođenja [algoritma] mora biti „razumno”
- primjeri
 - množenje se može svesti na ponavljanje zbrajanja – djelotvorno, ali nije učinkovito!
 - zbrajanje prvih n prirodnih brojeva izravno $(1 + 2 + \dots + n)$ ili korištenjem Gaussove formule $n*(n + 1)/2$

Svojstva algoritma - učinkovitost

- algoritam koji bi izabirao potez igrača šaha tako da ispita sve moguće posljedice poteza, zahtijevao bi milijarde godina na najbržem zamislivom računalu



- broj mogućih poteza je procijenjen na oko $12 \cdot 10^{81}$ (A.S. Fraenkel, 2007.)

Podrijetlo riječi algoritam

- iz srlat. *algorithmus*, *algorismus* (transliteracija arap. *Al-Khowārizmi*)
 - stariji oblik starofranc. *algorisme* („arapski brojevni sustav”, 13. st.)
iz srlat. *algorismus*, transliteracija al-Khwarizmi (rodom iz Khwarizma)
- prema imenu perzijskog matematičara Abu Ja'far Mohammed ibn Mûsâ al'Khowârizmî (~780 - ~850)

al'Khowârizmî

- perzijski matematičar Abu Ja'far Mohammed ibn Musa al Khowarizmi (Khwarizmi, ~780 – Bagdad, ~850)
- potiče korištenje hindu-arapskih brojeva (H. Zemanek, 1979.)
- u Bagdadu oko 825. godine napisao knjigu “Hidab al-jabr w'al-muqubala” (u 12. st. prevedeno na latinski): Znanost o prenošenju i poništenju



Prvi algoritmi

- stari vijek
 - prvo su se pojavili *recepti* za kuhanje, obavljanje poljoprivrednih poslova, itd.
 - algoritmi za faktORIZACIJU i određivanje drugog korijena (Babilon, ~1600. g. pr. Kr.)
 - Euklidov algoritam (~300. g. pr. Kr.); Eratostenovo sito (~200. g. pr. Kr.)
 - stari Grci su preuzeli mnoštvo postupaka od Babilonaca, Egipćana i dr.
- u srednjovjekovnim rukopisima: „*Dixit Algorizmi*”
 - za postupke koje je opisao al'Khowârizmî; bili su pojam preciznosti i točnosti
 - algoritmi u kojima se koriste hindu-arapski brojevi – npr. za rješavanje linearnih i kvadratnih jednadžbi
- u početku algoritmom se nazivaju samo pravila računanja s brojevima, kasnije i pravila obavljanja ostalih zadataka u matematici

Algoritmi i strukture podataka (1)

- **programi** = algoritmi + strukture podataka
 - „Programs, after all, are concrete formulations of abstract algorithms based on particular representations and structures of data” (N. Wirth, 1985.)
- kako **osmisлити** algoritme?
- kako **strukturirati** podatke?
- kako **formulirati** algoritme?
- kako **verificirati** korektnost algoritama?
- kako **analizirati** algoritme?
- kako **provjeriti** (testirati) program?

Algoritmi i strukture podataka (2)

- postupci izrade algoritama nisu jednoznačni te zahtijevaju i kreativnost - inače bi već postojali generatori algoritama
 - znači da se (za sada?) gradivo ovog predmeta ne može u potpunosti algoritmizirati
 - „Poznavanje algoritama je karakteristika koja razlikuje vješte programere od početnika” (Cormen *et al.* 2009)
„Having a solid base of algorithmic knowledge and technique is one characteristic that separates the truly skilled programmers from the novices”

Algoritmi i programi

- **program** - opis algoritma koji u nekom programskom jeziku jednoznačno određuje što računalo treba napraviti
- **Na ovom predmetu**
 - koristit će se **programski jezik C++**
 - objedinjuje brzinu C-a i snagu OOP-a