Adatszerkezetek és algoritmusok

Bevezetés

Dr. Fazekas Attila

A tananyag elkészítését az EFOP-3.4.3-16-2016-00021 számú projekt támogatta. A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósult meg.

DEBRECENI EGYETEM

Az algoritmus fogalma

- **Algoritmus**nak nevezünk bármilyen **jól definiált** számítási eljárást, amely **bemenet**ként véges számú értéket kap és **kimenet**ként véges számú értéket állít elő **véges számú lépés**ben.
- Az algoritmusok gyakran olyan eszközök, amelyek segítségével pontosan meghatározott számítási feladatokat oldunk meg.
- Egy algoritmust **helyes**nek mondunk, ha minden bemenetre megáll és helyes eredményt ad.

DEBRECENI
EGYETEM

Néhány példa algoritmusra

- Human Genom Project célja az emberi DNS térkép elkészítése, amely során 100 000 gént, 3 milliárd kémiai bázispárt kell meghatározni.
- Internet esetén a jó utak megtalálása az adatok továbbításához.
- Nyilvános kulcsú titkosítás. RSA, PGP.
- Autós navigáció.
- Órarend elkészítése.

DEBRECENI
EGYETEM

Adatszerkezetek

- Az **adatszerkezet** adatok tárolására és szervezésére szolgáló eszköz, amely lehetővé teszi a hozzáférést és módosításokat.
- Egyetlen adatszerkezet sem megoldás minden célra.
- Fontos, hogy ismerjük az egyes adatszerkezetek **előny**eit és **hátrány**ait.
- Különböző programozási nyelvek az egyes adatszerkezeteket esetenként **eltérő**en valósítja meg. Esetenként bizonyos adatszerkezetek **hiányozhat**nak az adott nyelvben.

DEBRECENI EGYETEM

Nehéz „algoritmusok”

- Egy **algoritmus hatékonysága** alatt a „sebességét” értjük.
- Vannak olyan feladatok, amelyekre nem ismerünk hatékony algoritmust, ezek az ún. **NP-teljes feladatok**.
- Ha egyetlen NP-teljes feladatra létezik hatékony algoritmus, akkor mindegyikre létezik!
- Gyakran találkozhatunk velük, és gyakran hasonlítanak olyan feladatokhoz, amelyekhez létezik hatékony algoritmus.

Négyszín-tétel

- Tekintsünk egy térképet, amit szeretnénk úgy kiszínezni, hogy a szomszédos országok más-más színűek legyenek.
- Könnyű látni, hogy három szín kevés.
- Nem túl nehéz belátni, hogy öt szín elég.
- A kérdés az, hogy **négy szín vajon elég-e?**
- Számítógépes programmal sikerült bizonyítani, hogy a kérdésre a válasz igen.
- Annak a meghatározása, hogy egy általános gráf kiszínezhető-e 4 színnel NP-teljes probléma.

DEBRECENI
EGYETEM

Lineáris keresés

- A legegyszerűbb **keresési** algoritmus, amely **rendezetlen tömbön** dolgozik.
- A tömb első elemétől kezdve összehasonlítjuk a tömbelemeket a keresendő elemmel. A keresési ciklus akkor ér véget, ha valamelyik tömbelem megegyezik a keresettel, vagy, ha a tömb végére érünk.
- Az algoritmus onnan kapta nevét, hogy a keresések száma, és így a futási idő, lineáris függvénye a tömb elemszámának.

DEBRECENI
EGYETEM

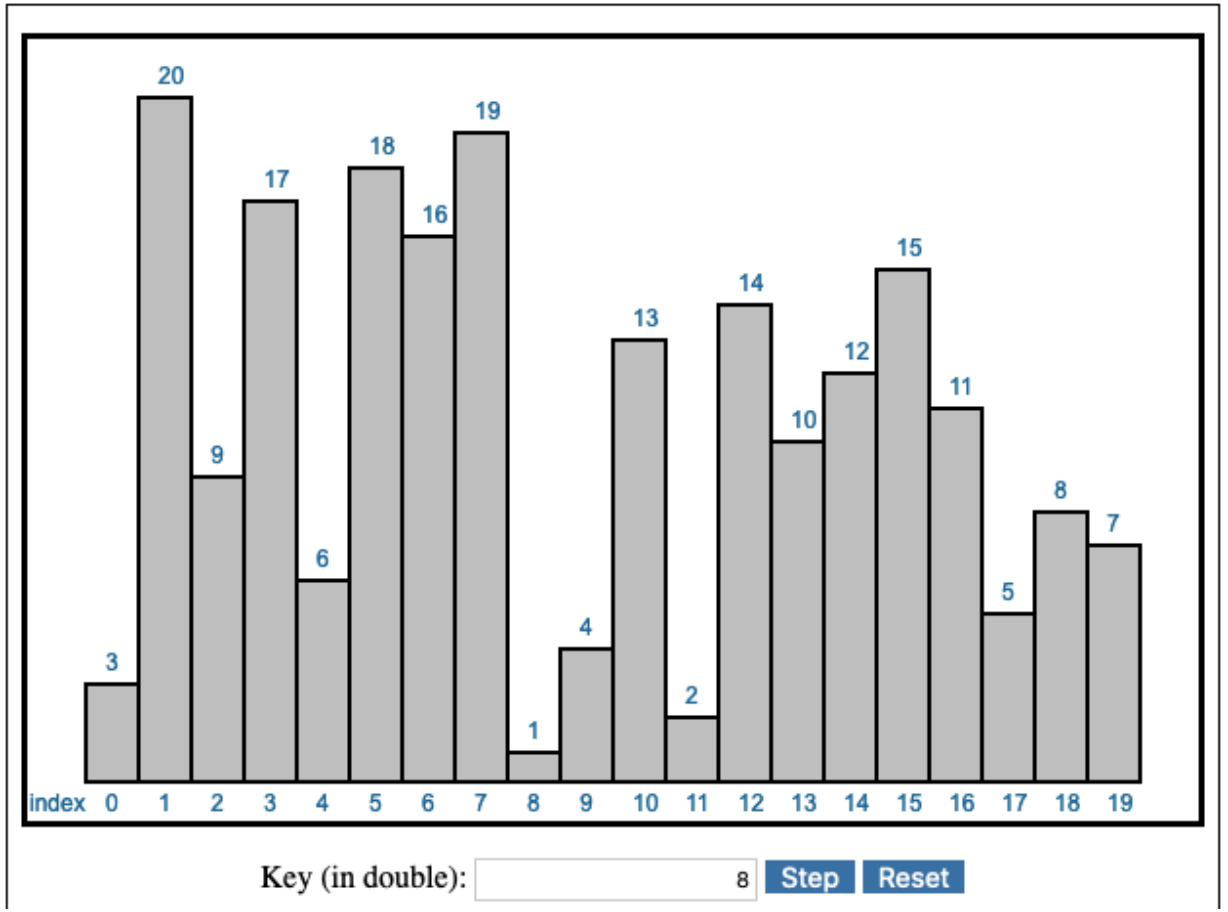
Mi is az a tömb?

A **tömb** (angolul array) olyan adatszerkezet amelyet nevesített elemek csoportja alkot, melyekre sorszámukkal (**index**ükkel) lehet hivatkozni. **Vektor**nak is nevezik, ha egydimenziós, **mátrix**nak esetenként, ha többdimenziós. A legtöbb programozási nyelvben minden egyes elemnek azonos adattípusa van és a tömb folytonosan helyezkedik el a számítógép memóriájában.

Forrás: Wikipédia

DEBRECENI EGYETEM

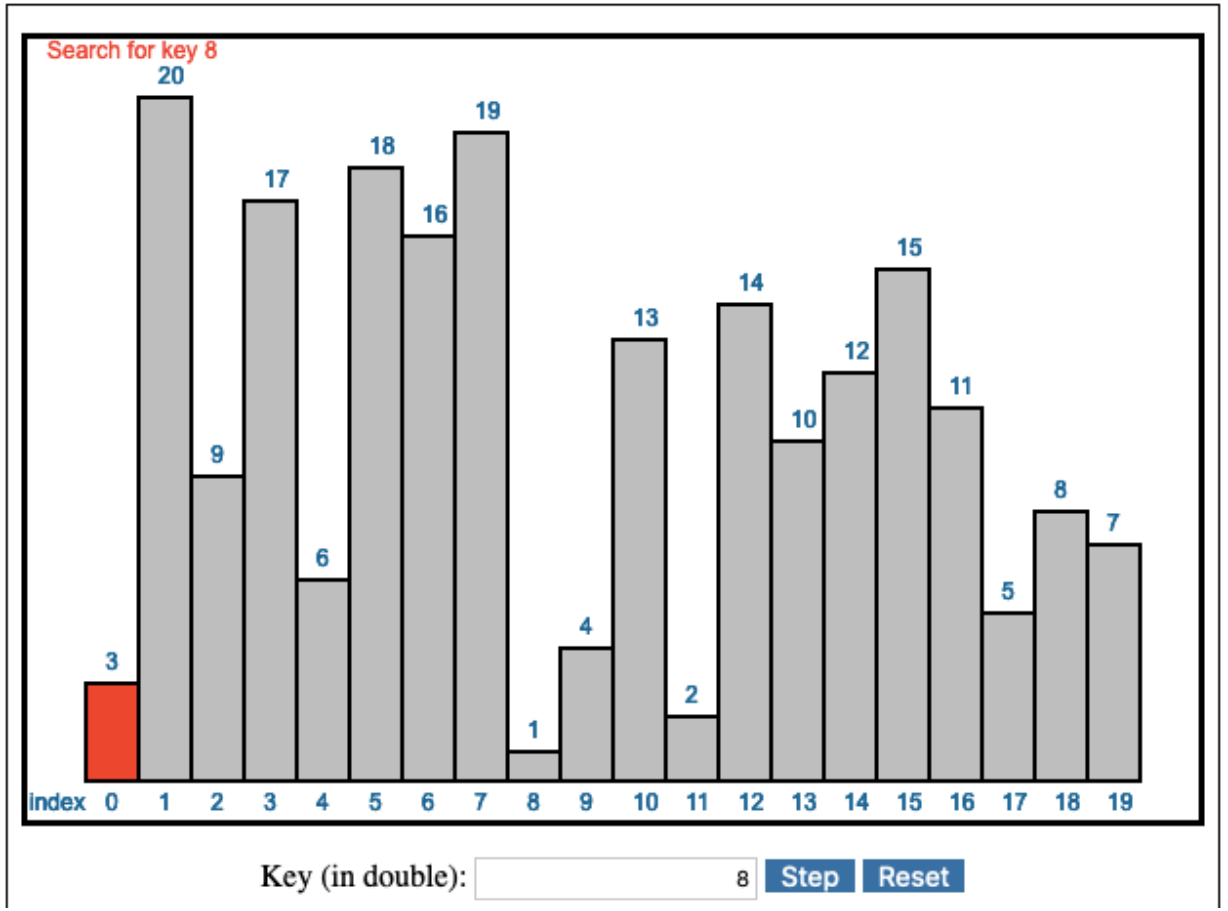
Lineáris keresés demonstrálása



DEBRECENI EGYETEM

Forrás: http://www.algoanim.ide.sk/index.php?page=showanim&id=73

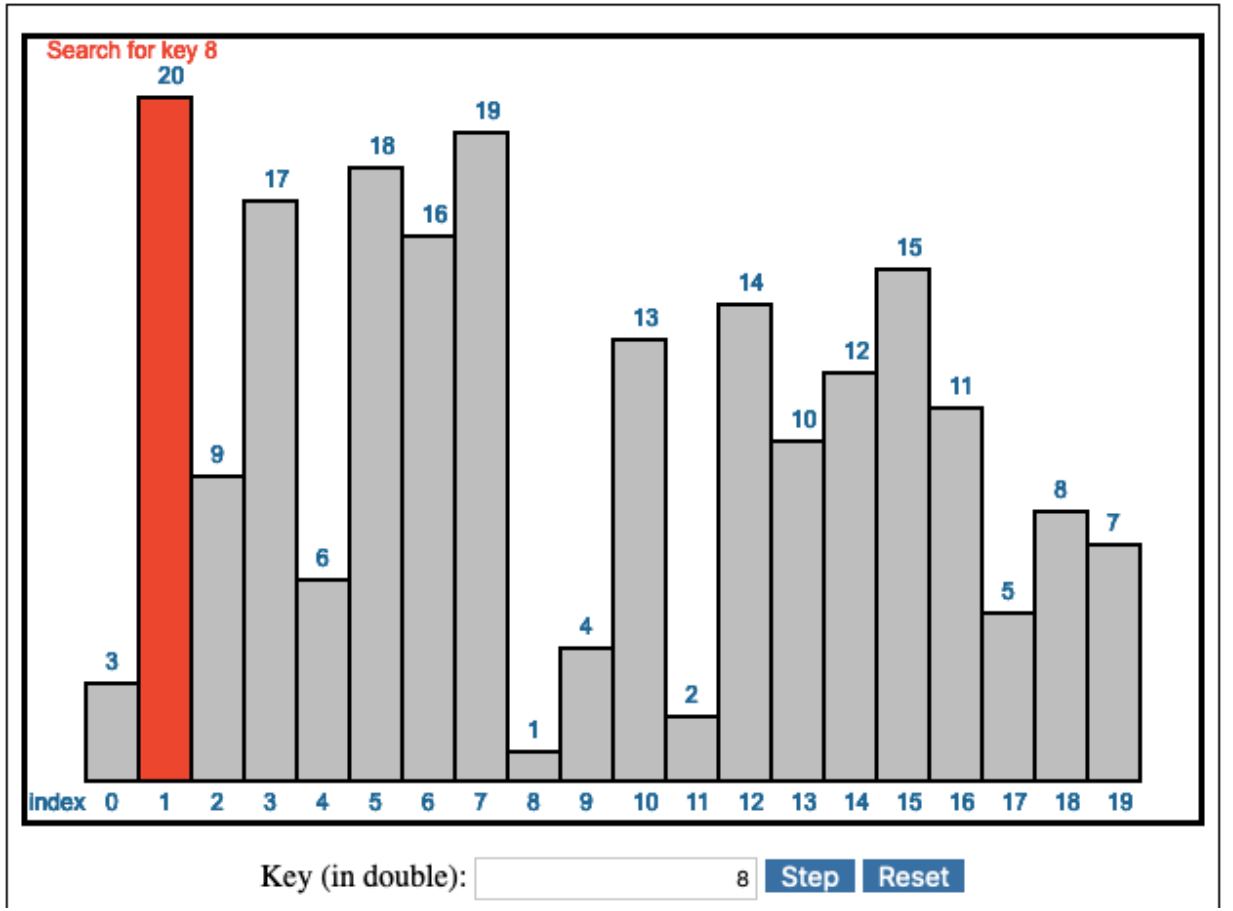
Lineáris keresés demonstrálása



DEBRECENI EGYETEM

Forrás: http://www.algoanim.ide.sk/index.php?page=showanim&id=73

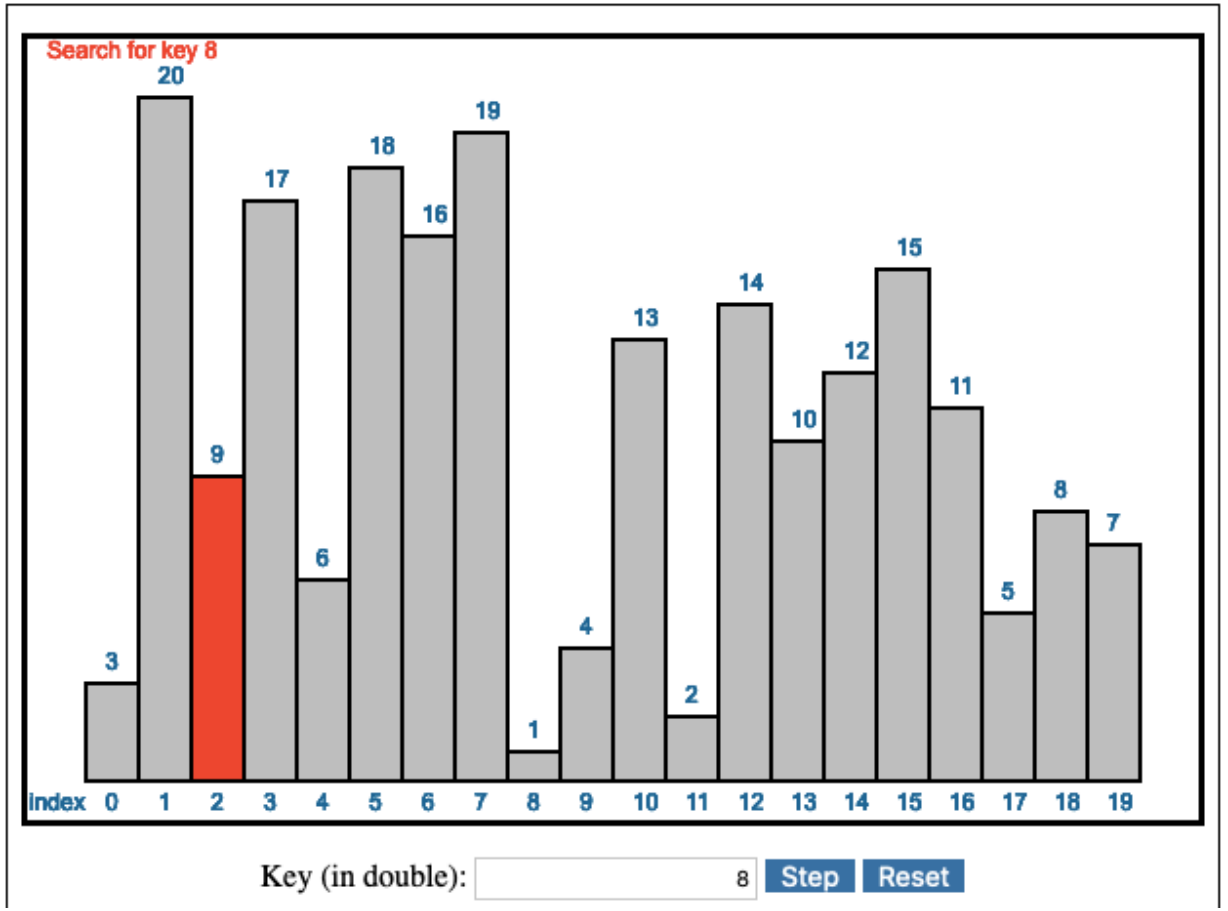
Lineáris keresés demonstrálása



DEBRECENI EGYETEM

Forrás: http://www.algoanim.ide.sk/index.php?page=showanim&id=73

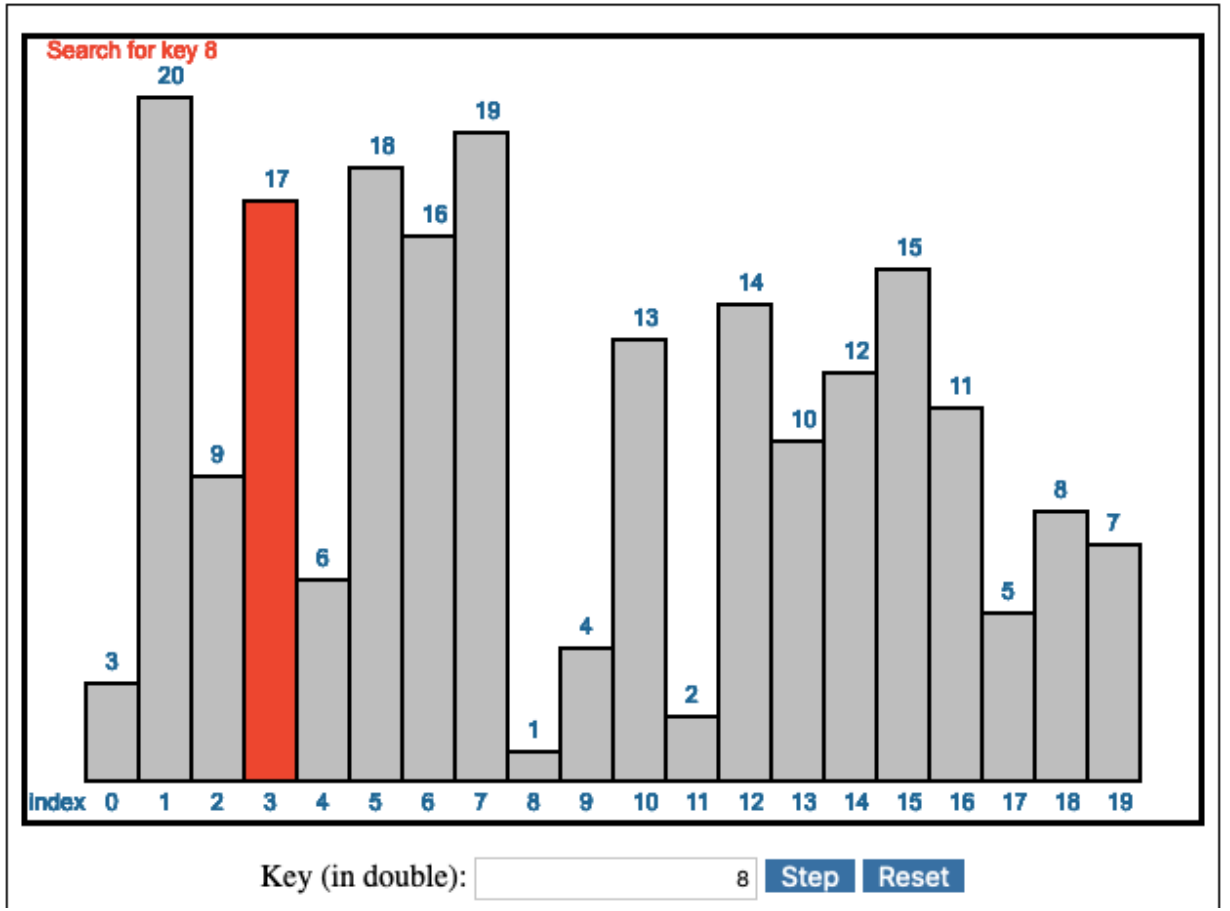
Lineáris keresés demonstrálása



DEBRECENI EGYETEM

Forrás: http://www.algoanim.ide.sk/index.php?page=showanim&id=73

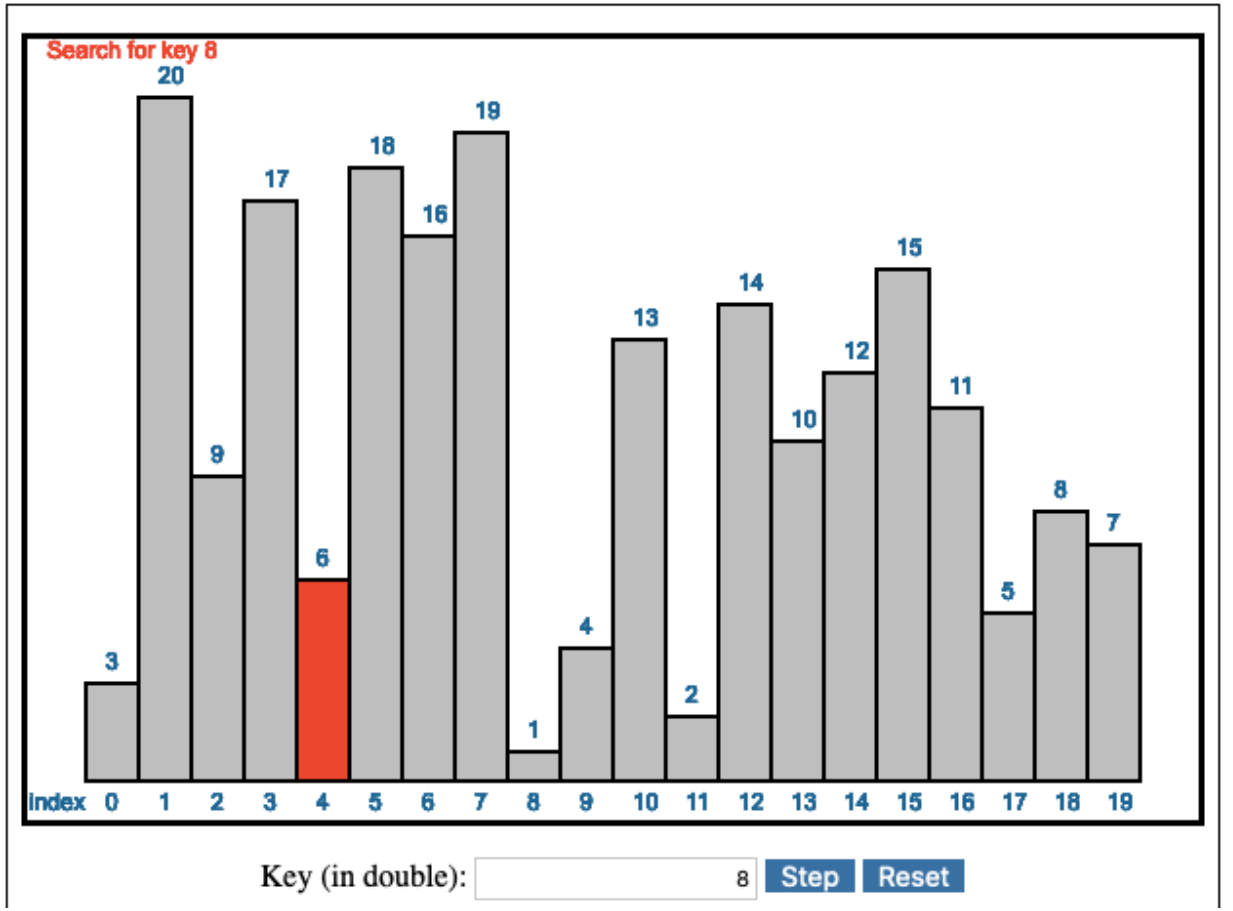
Lineáris keresés demonstrálása



DEBRECENI EGYETEM

Forrás: http://www.algoanim.ide.sk/index.php?page=showanim&id=73

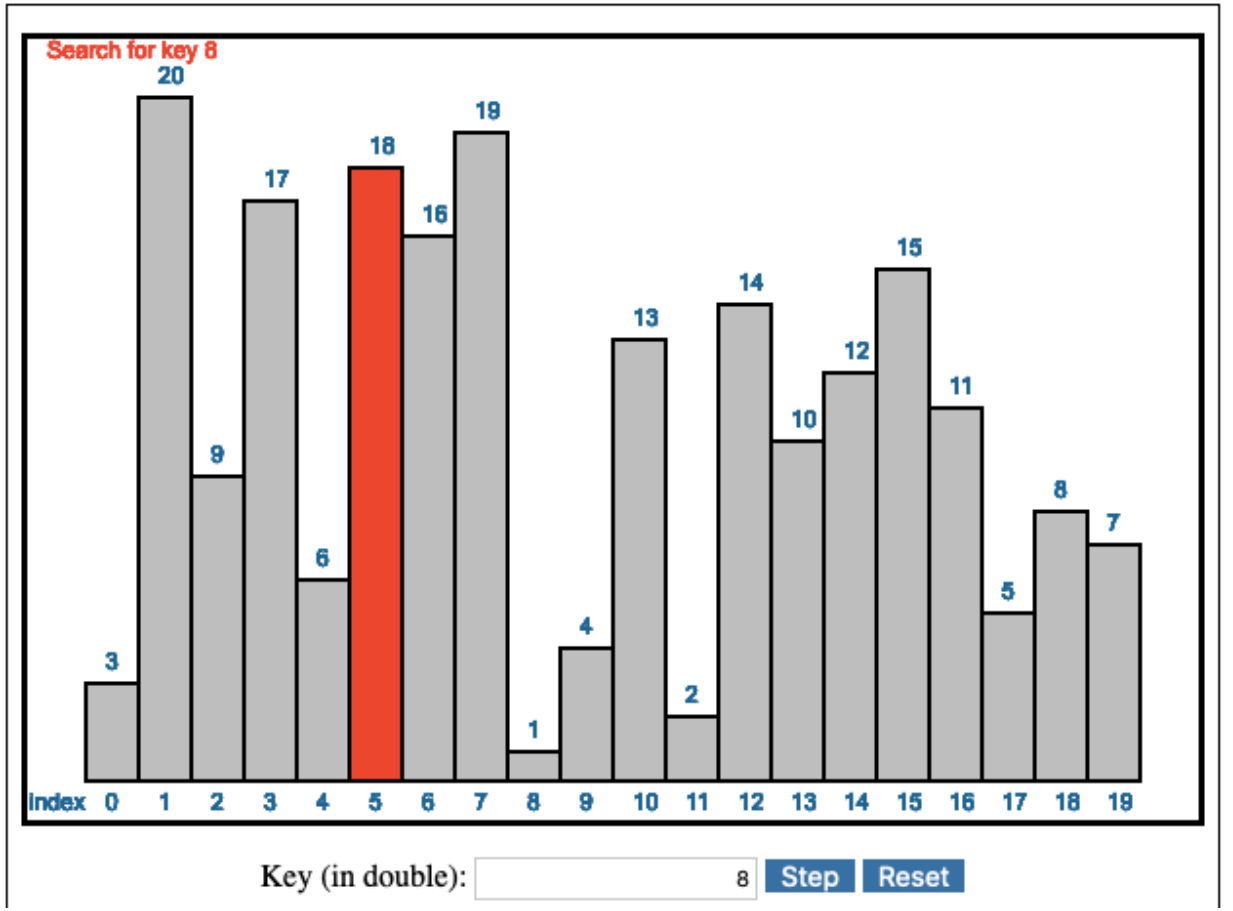
Lineáris keresés demonstrálása



DEBRECENI EGYETEM

Forrás: http://www.algoanim.ide.sk/index.php?page=showanim&id=73

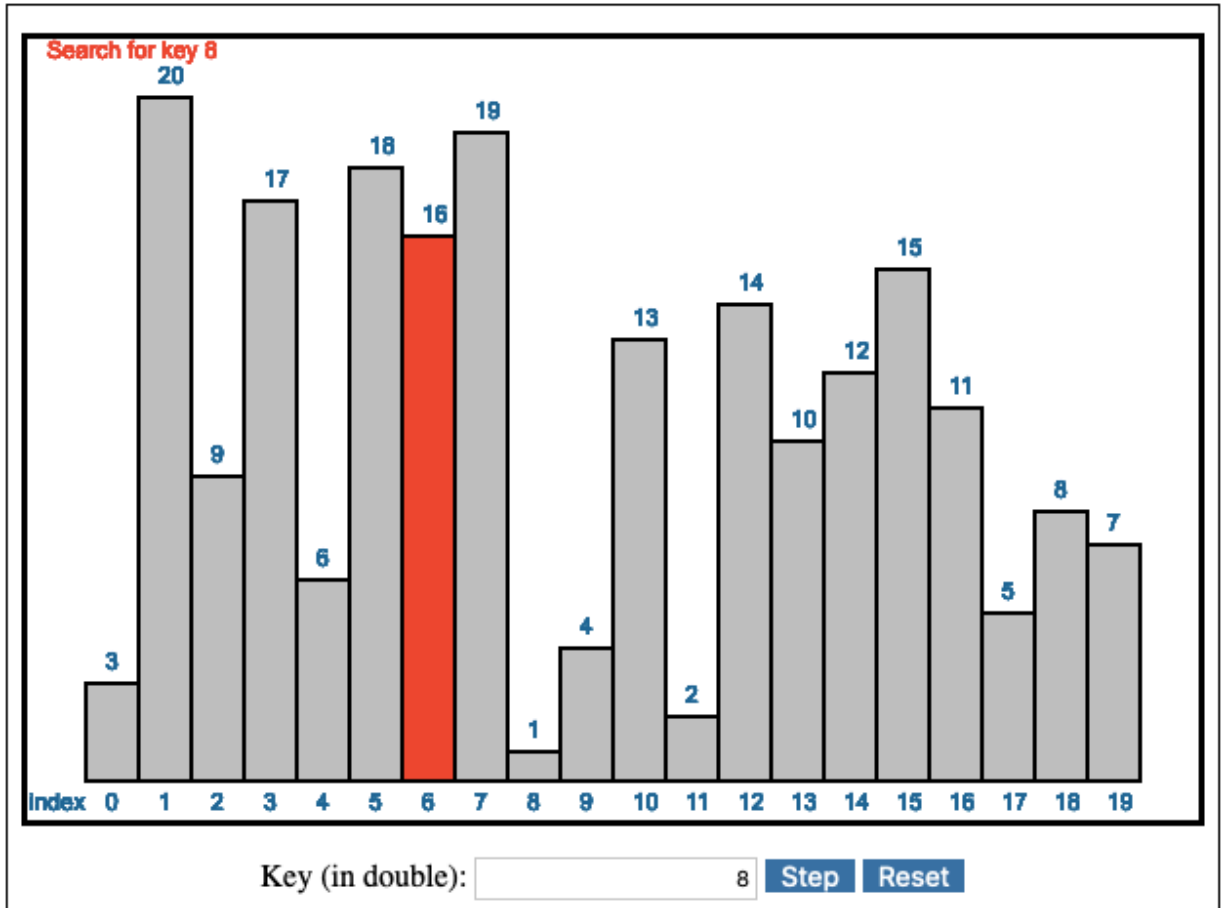
Lineáris keresés demonstrálása



DEBRECENI EGYETEM

Forrás: http://www.algoanim.ide.sk/index.php?page=showanim&id=73

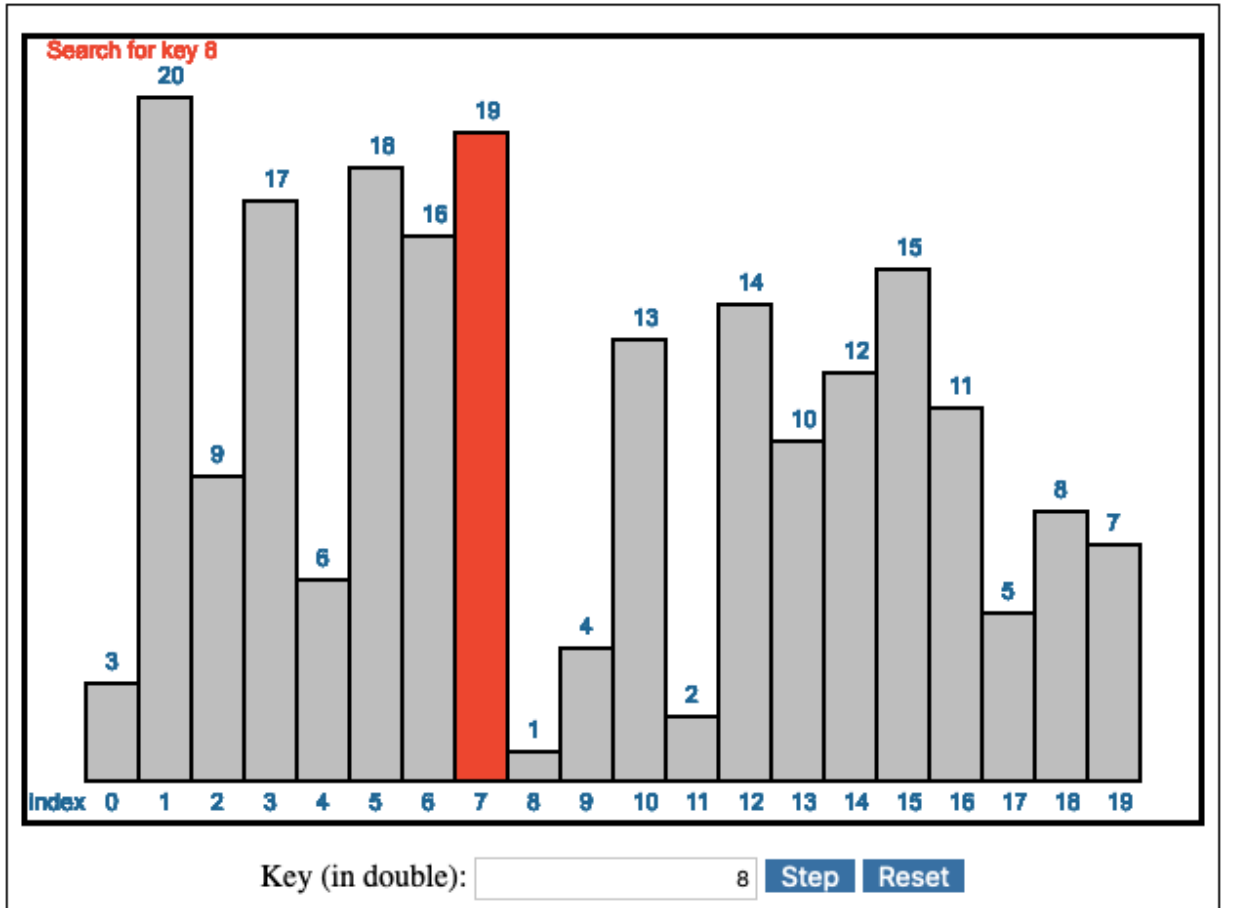
Lineáris keresés demonstrálása



DEBRECENI EGYETEM

Forrás: http://www.algoanim.ide.sk/index.php?page=showanim&id=73

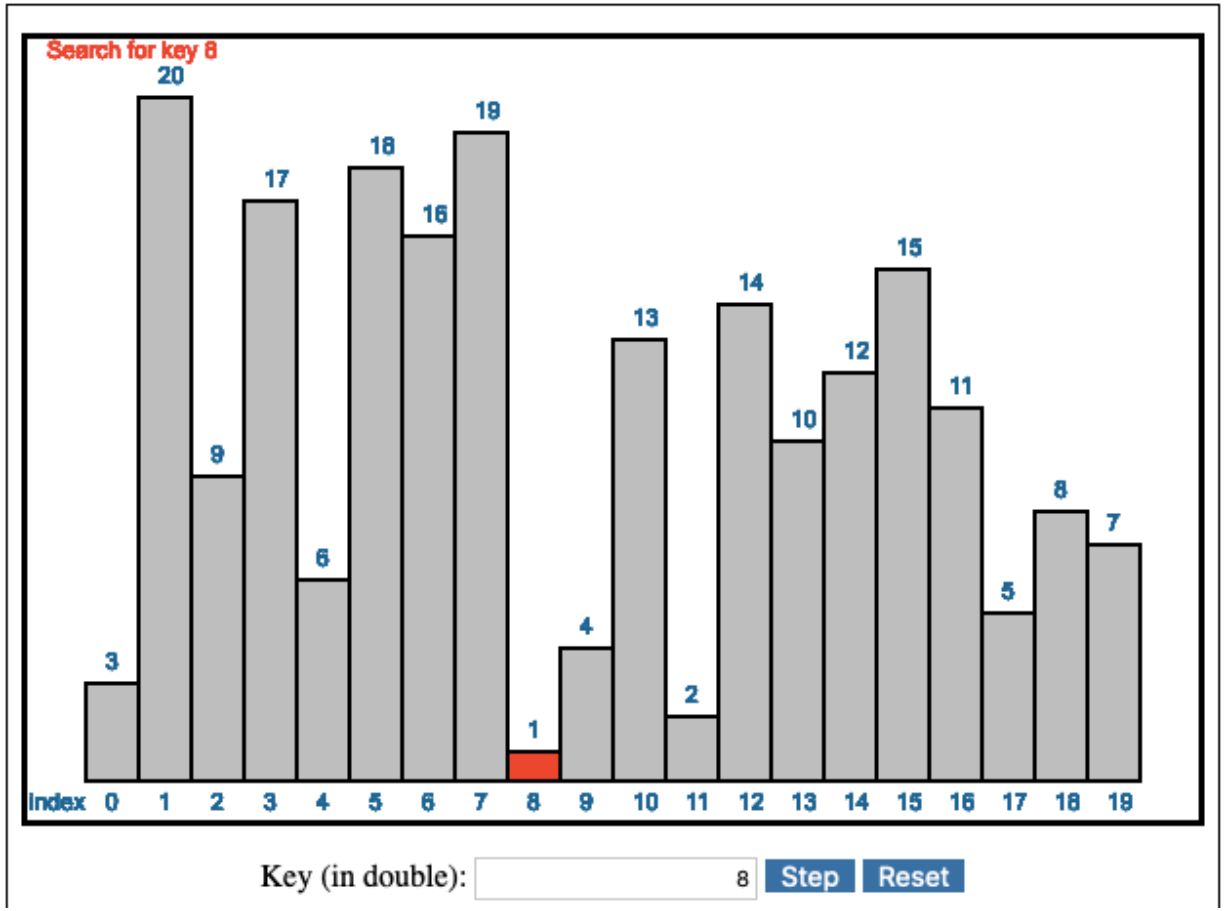
Lineáris keresés demonstrálása



DEBRECENI EGYETEM

Forrás: http://www.algoanim.ide.sk/index.php?page=showanim&id=73

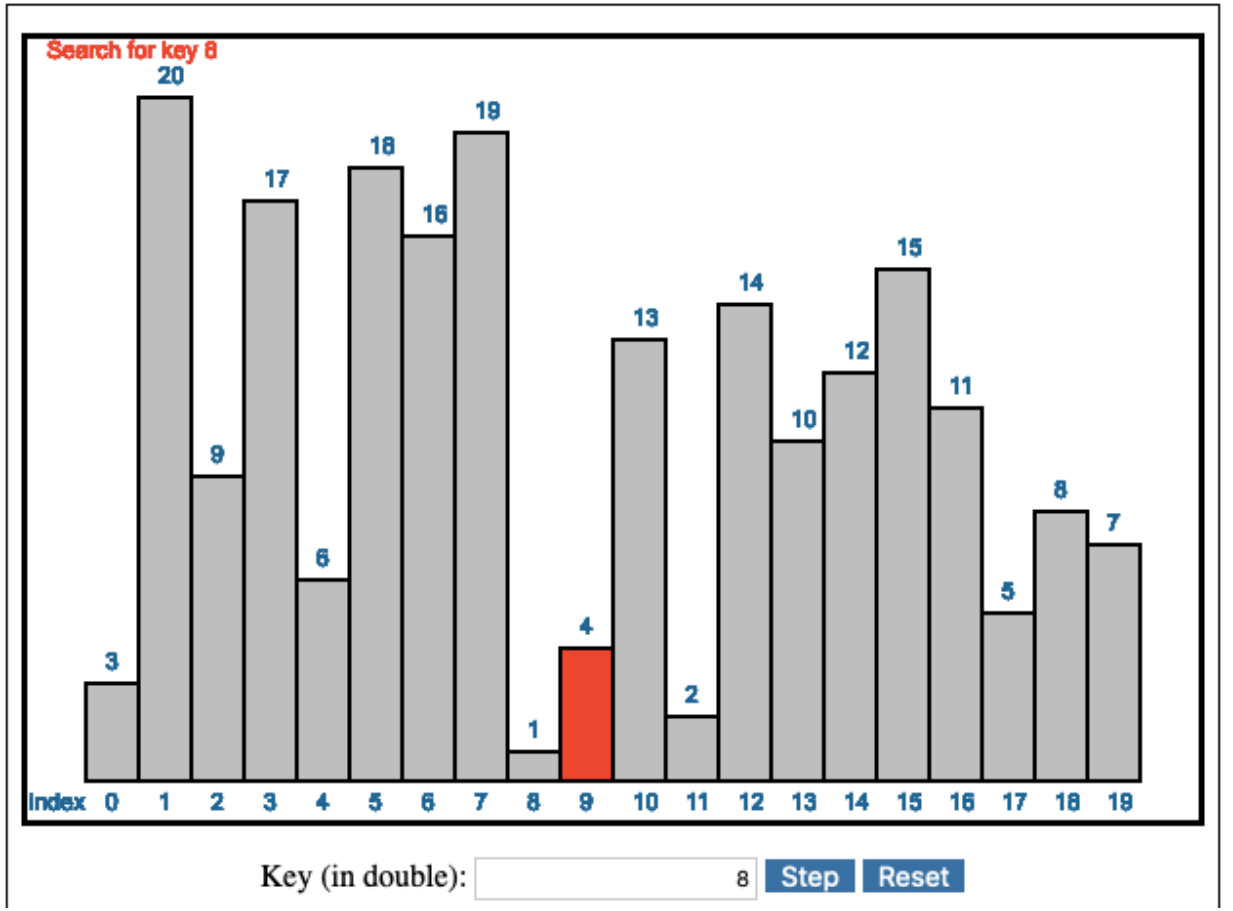
Lineáris keresés demonstrálása



DEBRECENI EGYETEM

Forrás: http://www.algoanim.ide.sk/index.php?page=showanim&id=73

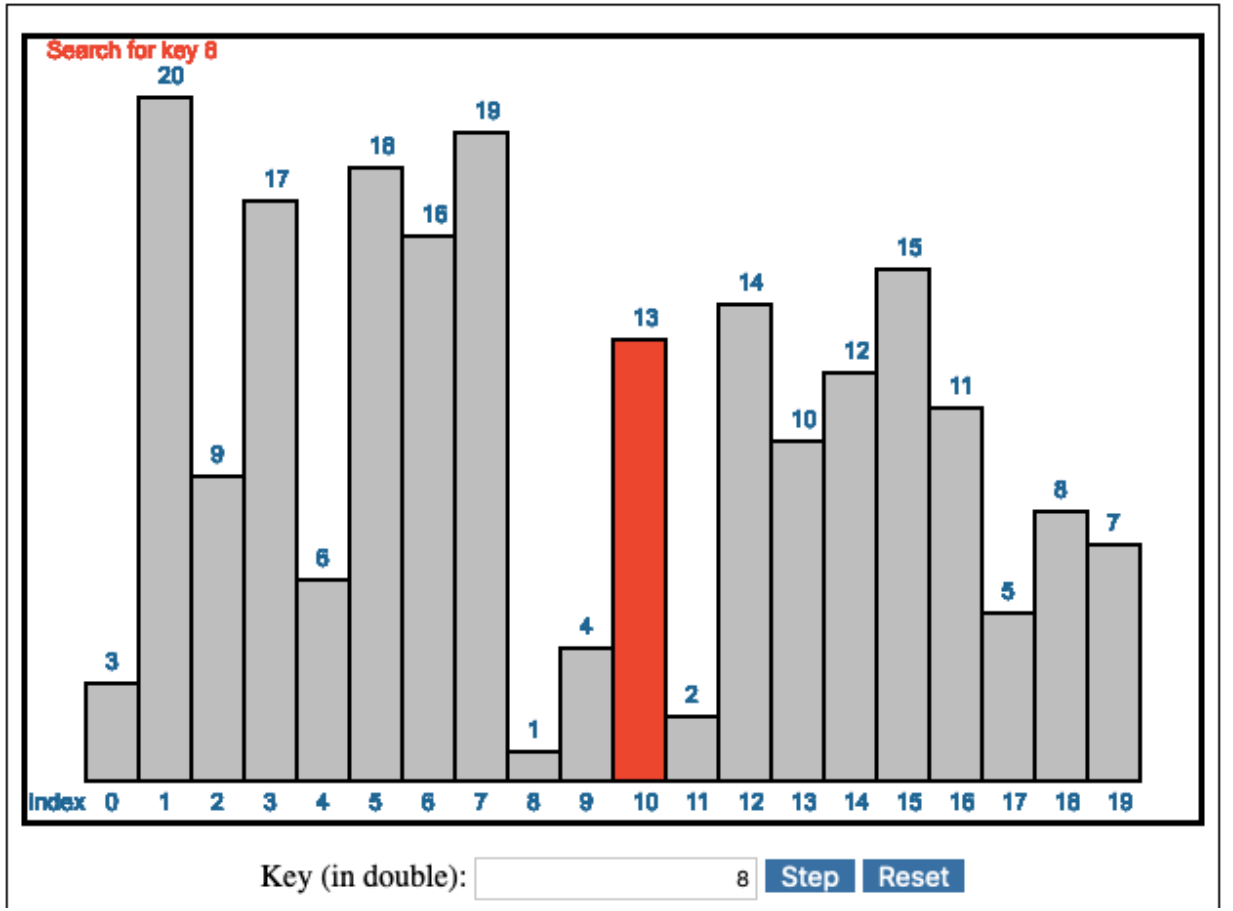
Lineáris keresés demonstrálása



DEBRECENI EGYETEM

Forrás: http://www.algoanim.ide.sk/index.php?page=showanim&id=73

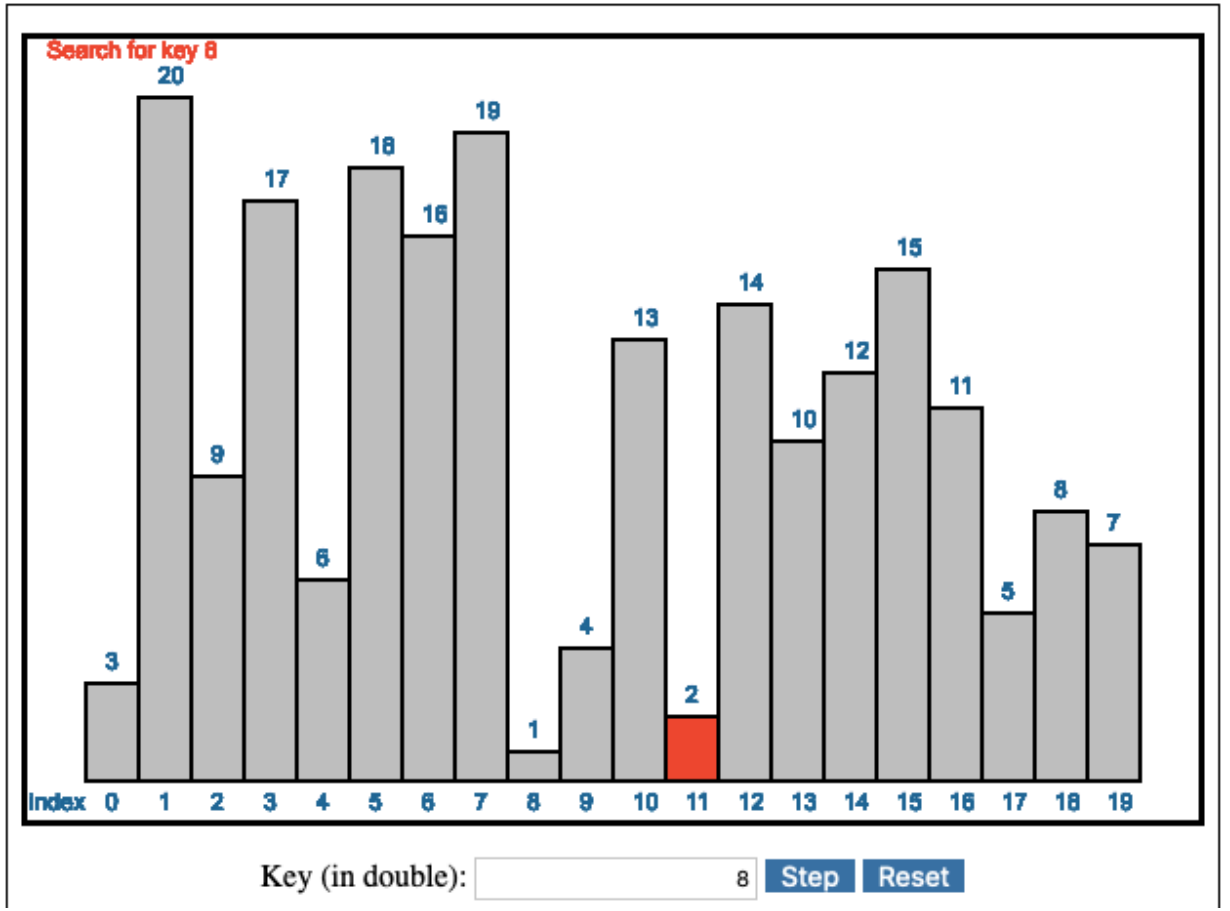
Lineáris keresés demonstrálása



DEBRECENI EGYETEM

Forrás: http://www.algoanim.ide.sk/index.php?page=showanim&id=73

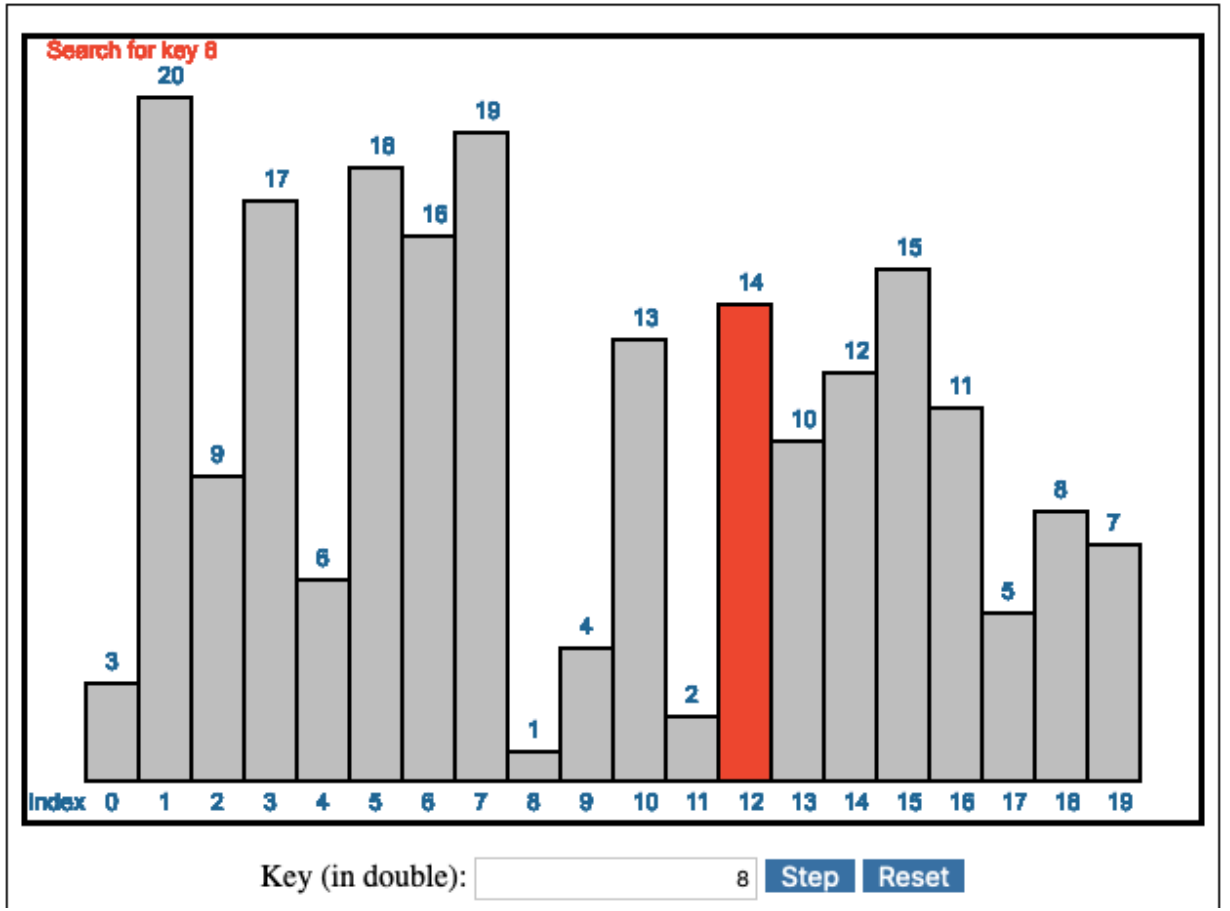
Lineáris keresés demonstrálása



DEBRECENI EGYETEM

Forrás: http://www.algoanim.ide.sk/index.php?page=showanim&id=73

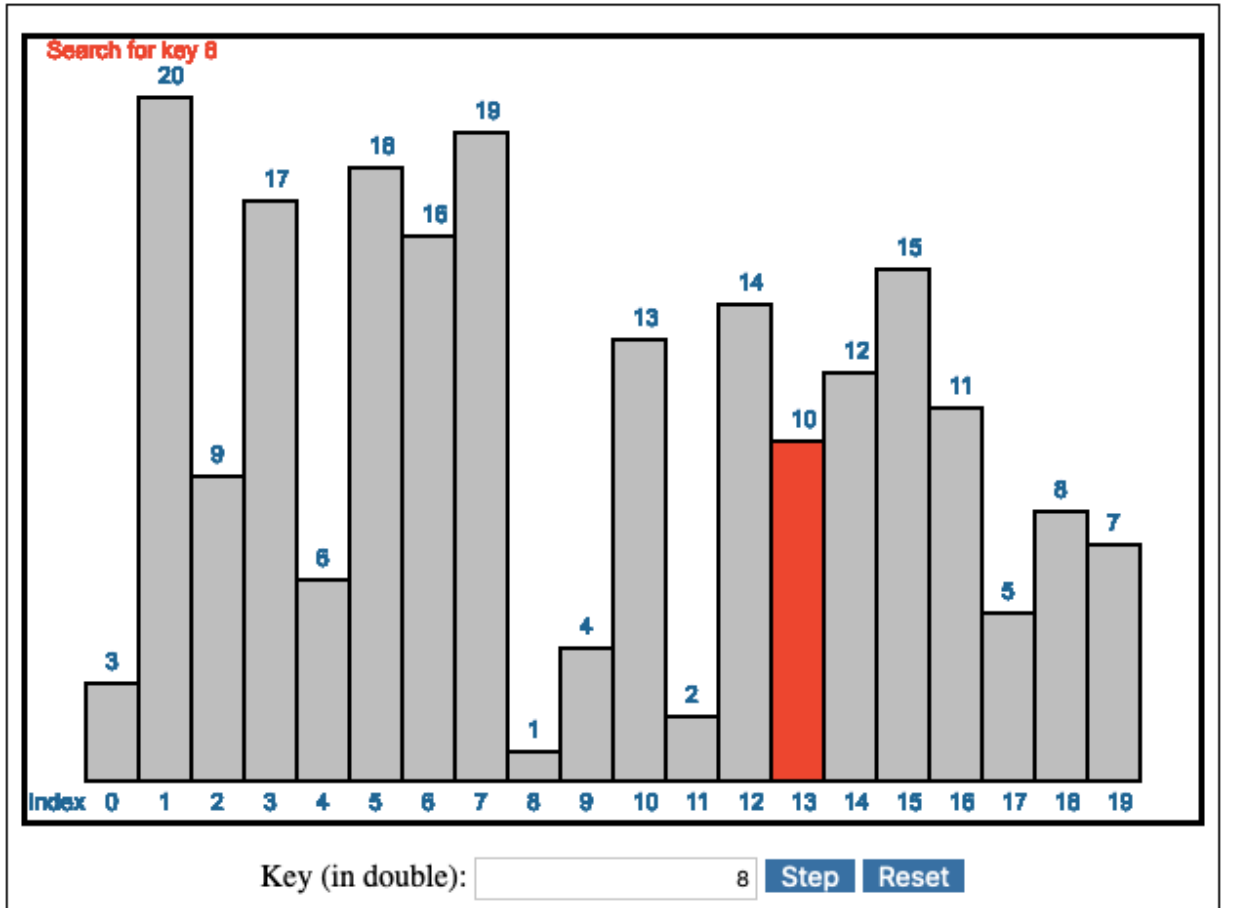
Lineáris keresés demonstrálása



DEBRECENI EGYETEM

Forrás: http://www.algoanim.ide.sk/index.php?page=showanim&id=73

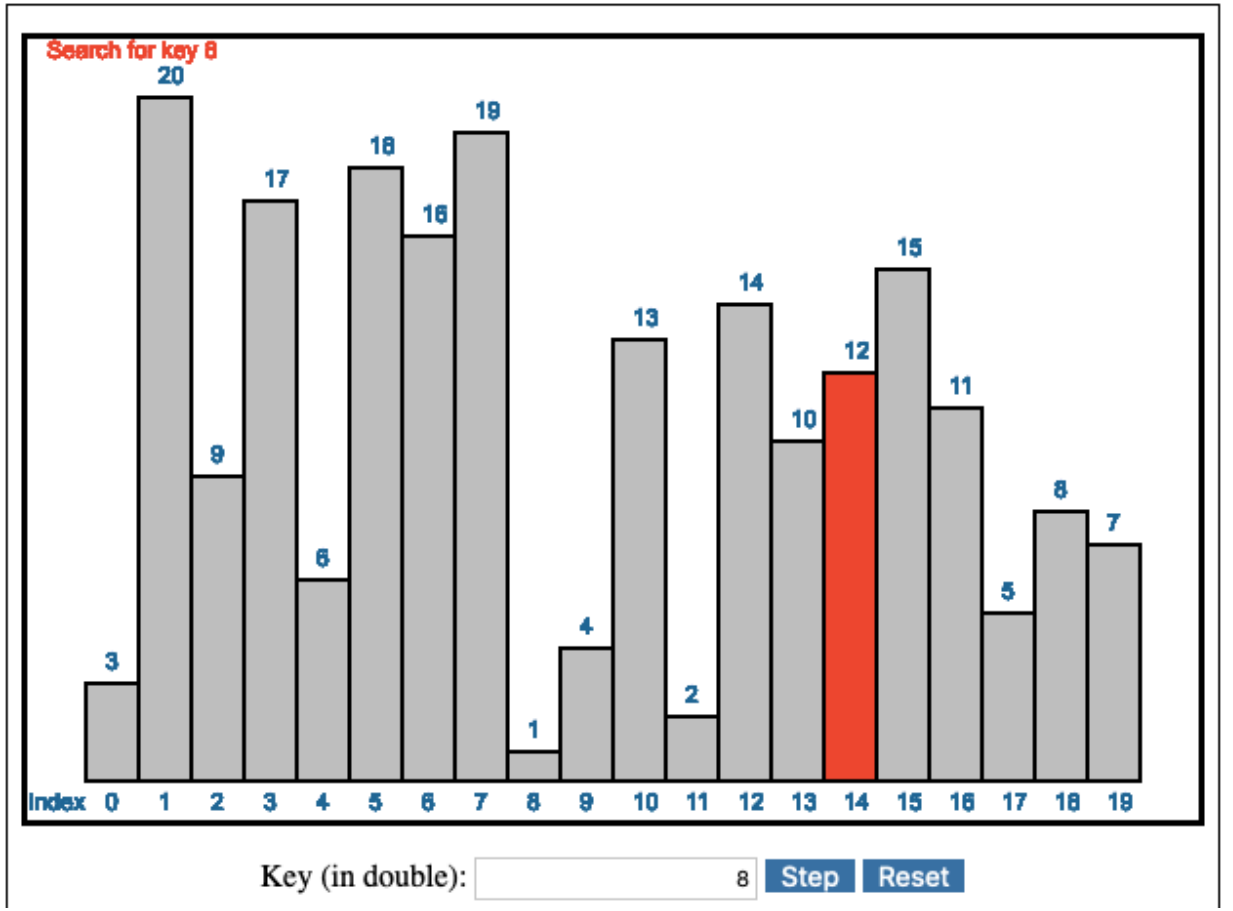
Lineáris keresés demonstrálása



DEBRECENI EGYETEM

Forrás: http://www.algoanim.ide.sk/index.php?page=showanim&id=73

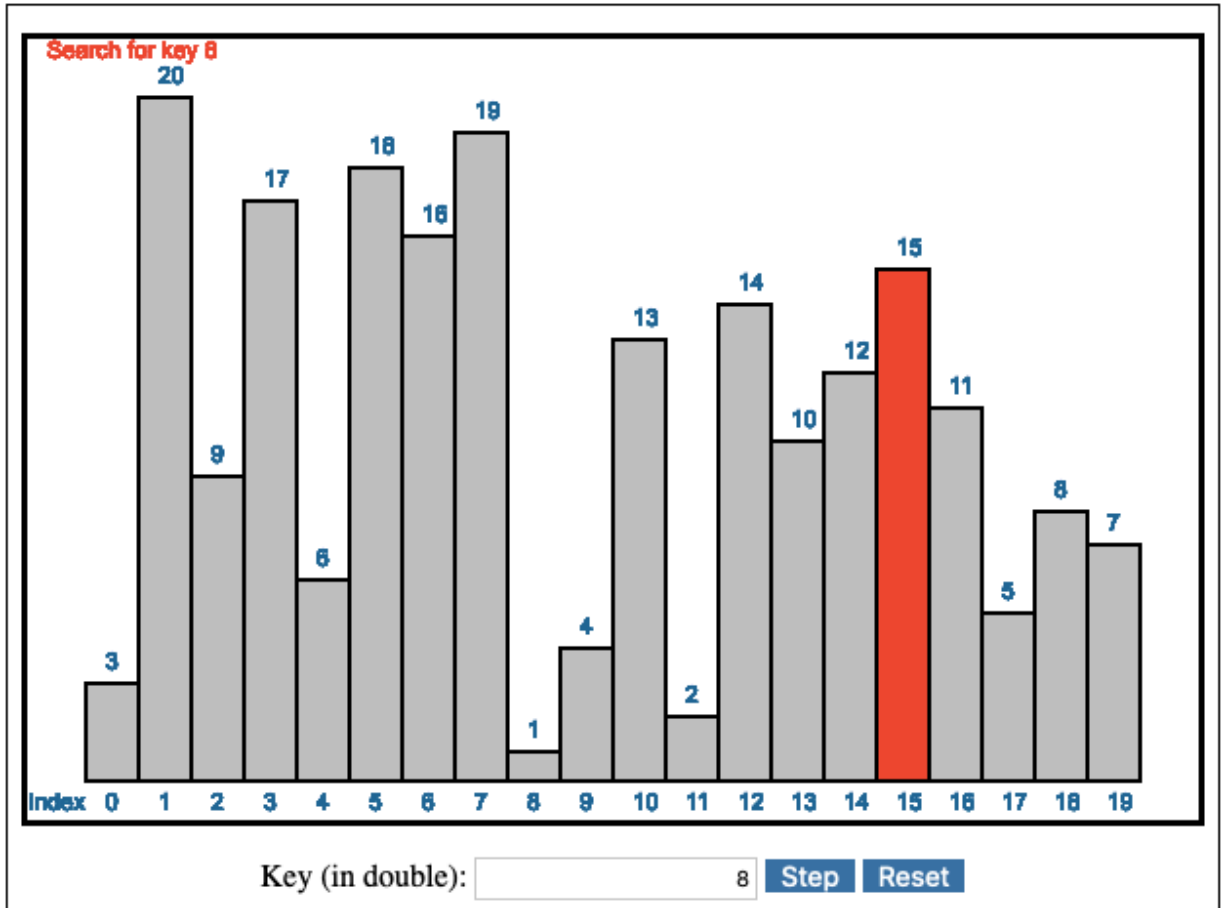
Lineáris keresés demonstrálása



DEBRECENI EGYETEM

Forrás: http://www.algoanim.ide.sk/index.php?page=showanim&id=73

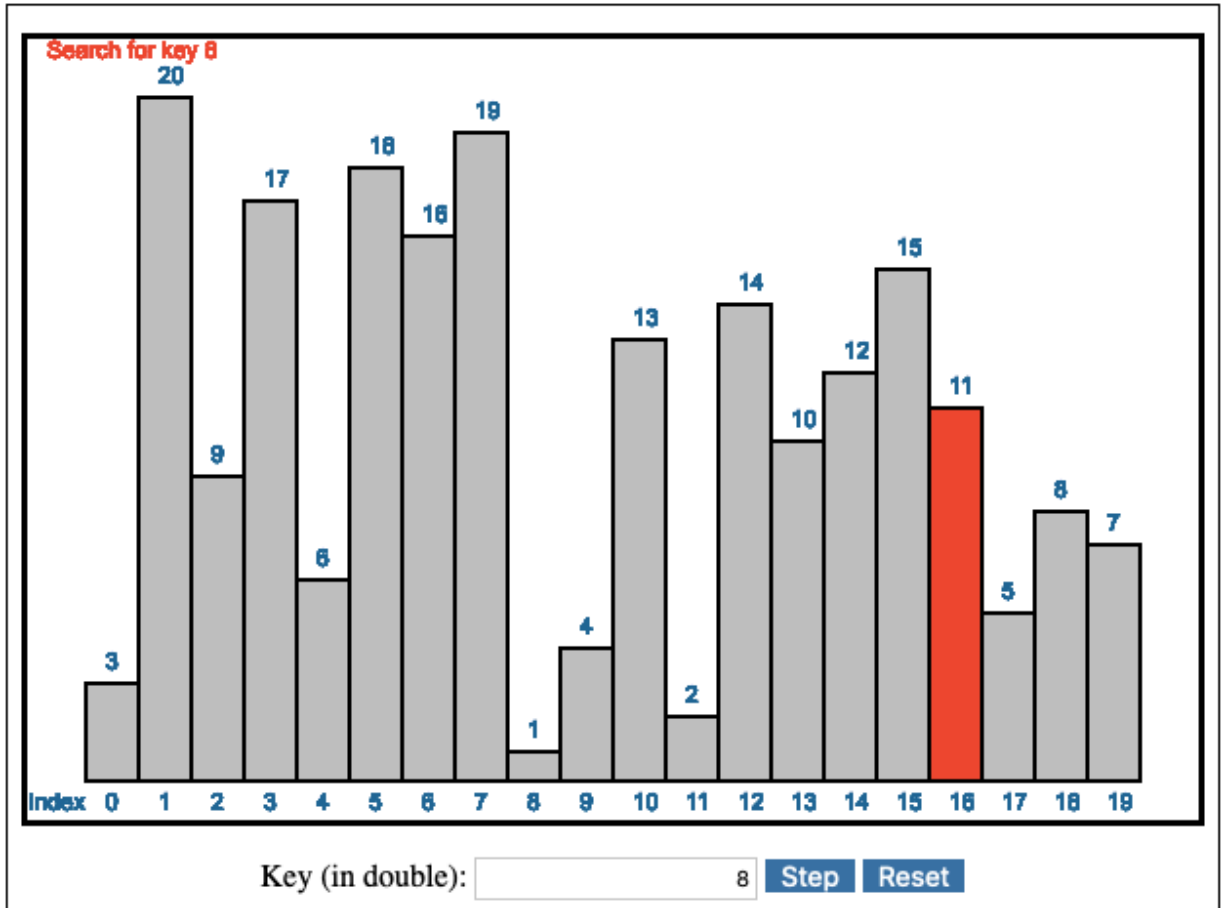
Lineáris keresés demonstrálása



DEBRECENI EGYETEM

Forrás: http://www.algoanim.ide.sk/index.php?page=showanim&id=73

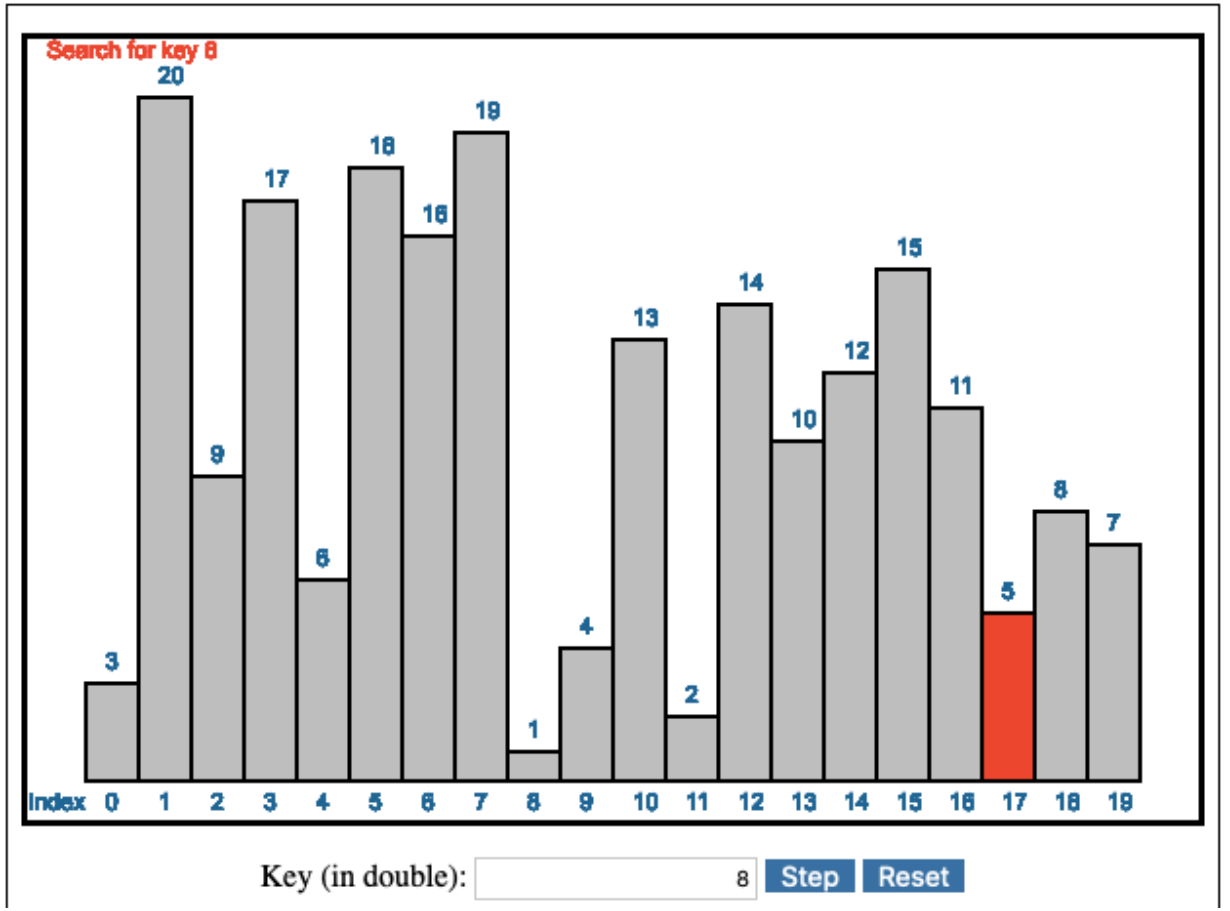
Lineáris keresés demonstrálása



DEBRECENI EGYETEM

Forrás: http://www.algoanim.ide.sk/index.php?page=showanim&id=73

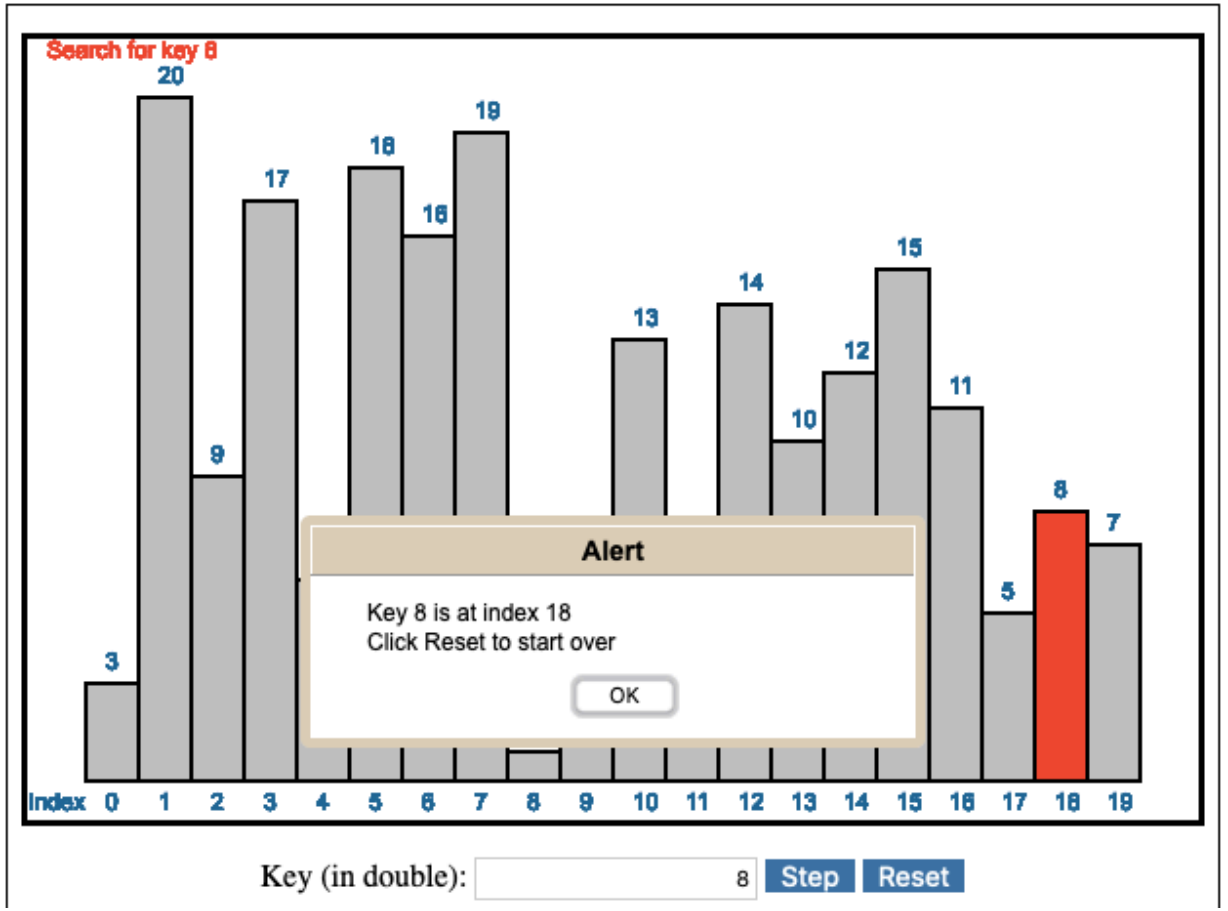
Lineáris keresés demonstrálása



DEBRECENI EGYETEM

Forrás: http://www.algoanim.ide.sk/index.php?page=showanim&id=73

Lineáris keresés demonstrálása



DEBRECENI EGYETEM

Forrás: http://www.algoanim.ide.sk/index.php?page=showanim&id=73

Lineáris keresés Pythonban

```
import numpy as np

def linearis_kereses(tomb, keresett):
    for i in range(len(tomb)):
        if tomb[i] == keresett:
            return i
    return -1

tomb=np.empty(20,np.int)
for i in range(len(tomb)):
    tomb[i]=i
print(linearis_kereses(tomb,8))
```

DEBRECENI EGYETEM

Lineáris keresés elemzése

- A bemenet mérete az elemek száma, amelyek között keresünk.
- Ha a keresett elem megtalálható a megadott elemek között, akkor a leghamarabb 1 lépésben (**legjobb eset**), legkésőbb utolsó lépésben (**legrosszabb eset**) találhatjuk meg.
- Ha a keresett elem nem található meg a megadott elemek között, akkor az utolsó hasonlítás után (n+1). lépésben ér véget az algoritmus.
- A bemenet növelése lineárisan növeli a lépések várható számát.

DEBRECENI
EGYETEM

Egy algoritmus elemzése

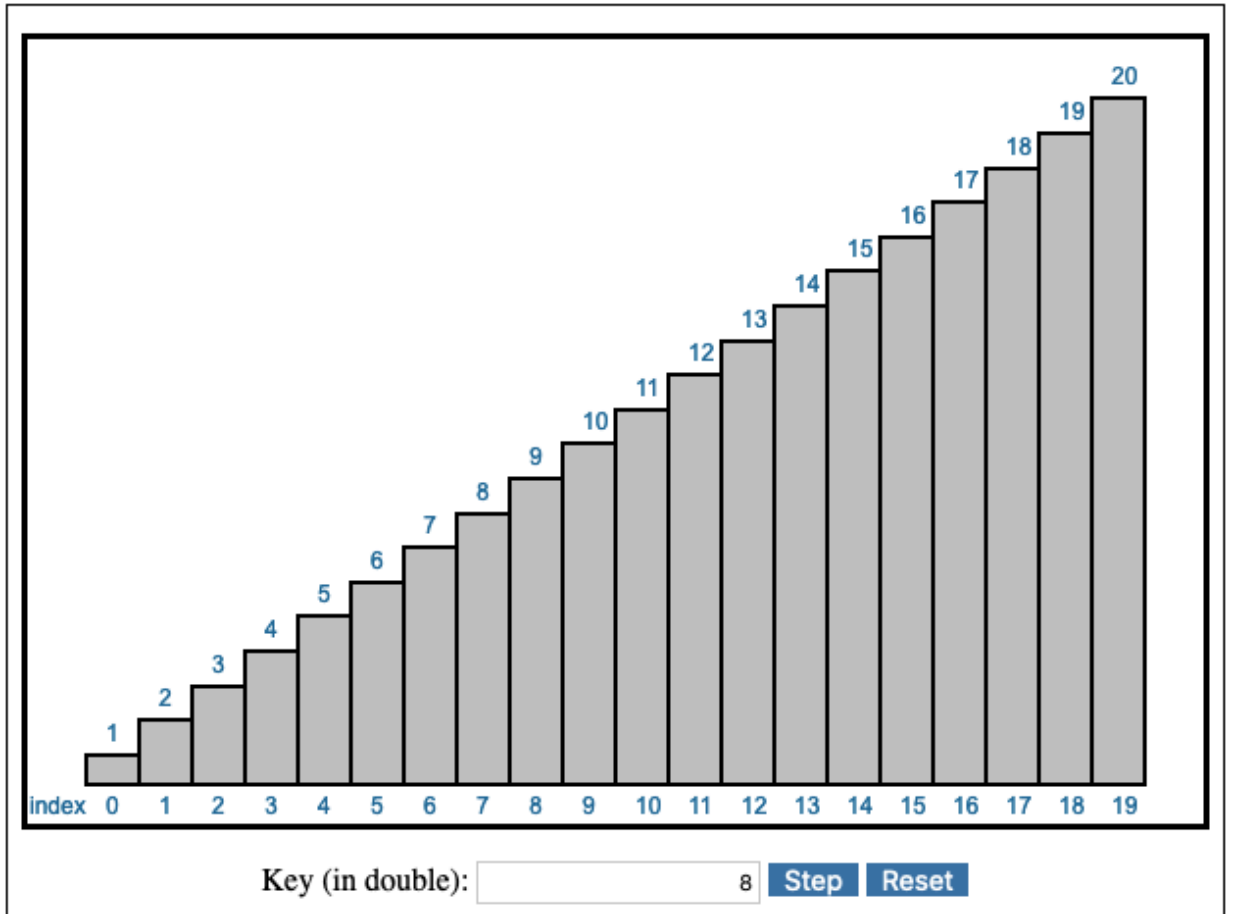
- Elemzés az algoritmus végrehajtásához szükséges erőforrások (memória, számítási idő, stb.) meghatározása.
- A **bemenet mérete** általában a bemenő elemek száma.
- Az **algoritmus futási ideje** a végrehajtott alapműveletek (“lépések”) száma. Tipikus alapműveletek: hasonlítás, értékadás, stb.

DEBRECENI
EGYETEM

Bináris keresés

- A **logaritmikus** vagy **bináris keresés**i algoritmus **rendezett adat**okon működik.
- A keresendő elemet először a tömb középső eleméhez hasonlítjuk. Ha a két elem egyezik, megtaláltuk a tömbelemet, ha nem, a keresést abban a tömbfélben folytatjuk, amelyet a középső és a keresett elem viszonya kijelöl.
- A keresést akkor fejezzük be, ha megtaláltuk a keresett tömbelemet, vagy a tömbrész elfogyott.

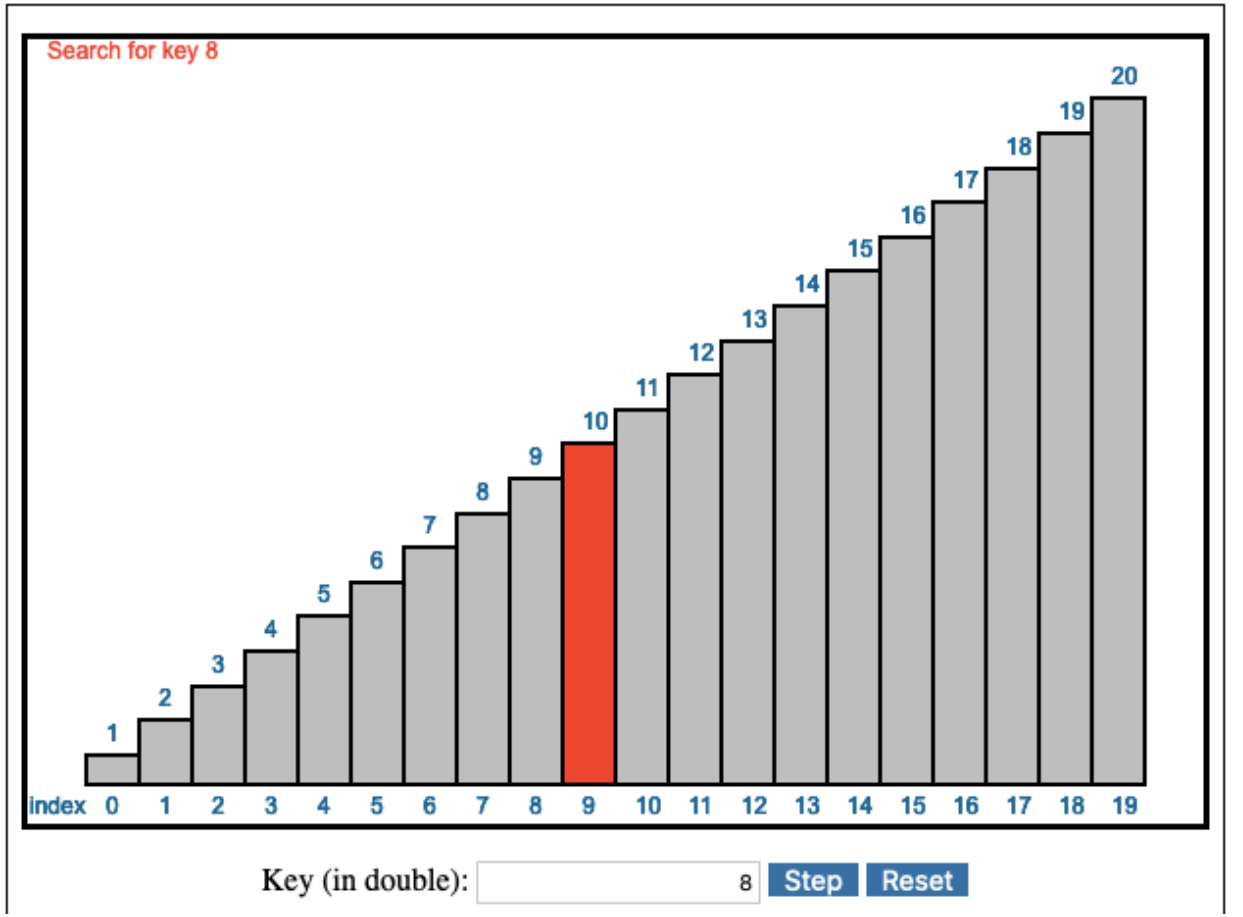
Bináris keresés demonstrálása



DEBRECENI EGYETEM

Forrás: http://www.algoanim.ide.sk/index.php?page=showanim&id=74

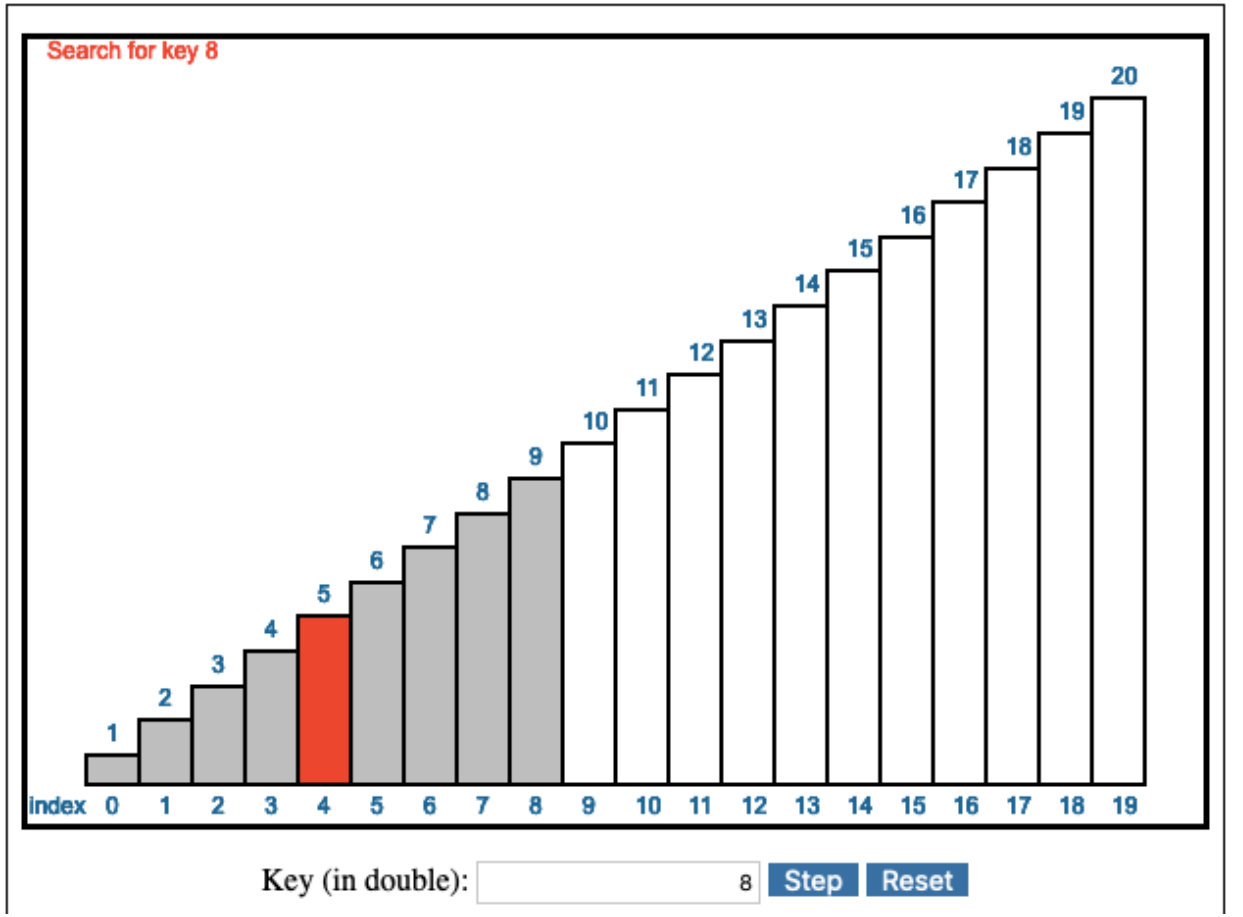
Bináris keresés demonstrálása



DEBRECENI EGYETEM

Forrás: http://www.algoanim.ide.sk/index.php?page=showanim&id=74

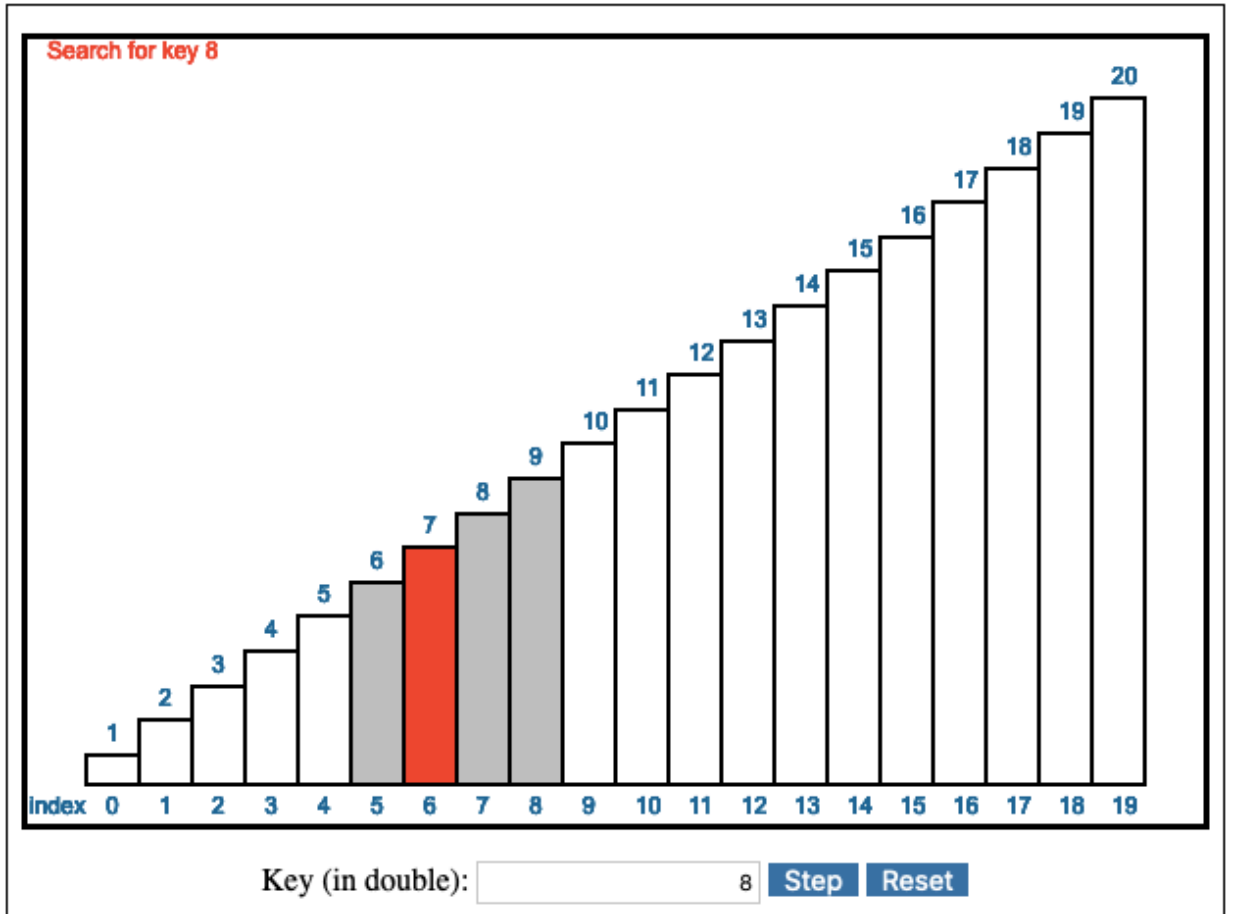
Bináris keresés demonstrálása



DEBRECENI EGYETEM

Forrás: http://www.algoanim.ide.sk/index.php?page=showanim&id=74

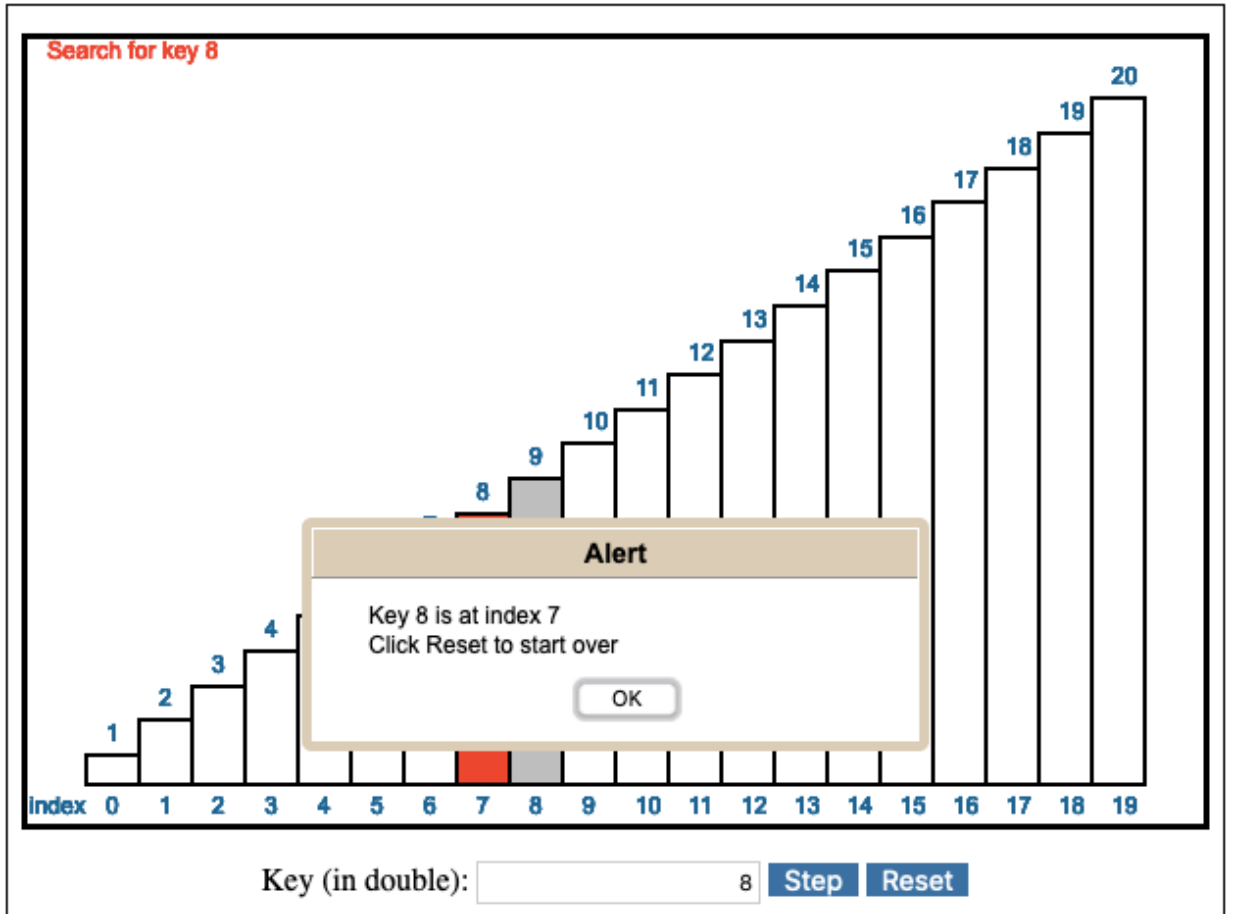
Bináris keresés demonstrálása



DEBRECENI EGYETEM

Forrás: http://www.algoanim.ide.sk/index.php?page=showanim&id=74

Bináris keresés demonstrálása



DEBRECENI EGYETEM

Forrás: http://www.algoanim.ide.sk/index.php?page=showanim&id=74

Bináris keresés Pythonban

```
def binaris_kereses(tomb, keresett):
    eleje=0
    vege=len(tomb)-1
    while eleje<=vege:
        i=(eleje+vege)//2
        print(eleje, vege)
        if tomb[i]==keresett:
            return i
        elif tomb[i]>keresett:
            vege=i-1
        elif tomb[i]<keresett:
            eleje=i+1
    return -1
```

DEBRECENI EGYETEM

Bináris keresés elemzése

- Leghamarabb az első lépésben találjuk meg az elemet.
- Legrosszabb esetben utolsóként találja meg. De ez mit is jelent?
- Tegyük fel, hogy 2^n darab elemünk van. Ekkor a lépések száma $\log(2^n)=n$ lesz. **Logaritmikus keresés**.
- Az adatok lineáris növekedése a műveletek számánál logaritmikus növekedését vonja maga utána.
- **Rendezett adatok között gyorsabb a keresés!**

DEBRECENI
EGYETEM

Köszönöm a figyelmet!