Wyszukiwanie binarne

Alicja Chmielewska
Adrianna Jajkowska
Antonina Brzezinka

Typy wyszukiwań

- Wyszukiwanie liniowe (sekwencyjne) - sprawdzany jest każdy element po kolei

- Wyszukiwanie binarne - unika sprawdzania każdego elementu zbioru

Wyszukiwanie binarne

Ogólnie rzecz biorąc: szybciej, lepiej, skuteczniej

Jak to działa?

Dany jest zbiór X i szukana k = 2

X= {1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17}

(1 + 17) / 2 = 9

k = 9 ?

(Nie)

k > 9 czy k < 9 ?

k < 9

{1 2 3 4 5 6 7 8}

(1 + 8) / 2 = 4,5 (4)

k = 4 ?

(Nie)

k > 4 czy k < 4 ?

k < 4

{1 2 3}

(1 + 3) / 2 = 2

k = 2?


TAK!

Jak sprawdzić liczbę potrzebnych kroków?

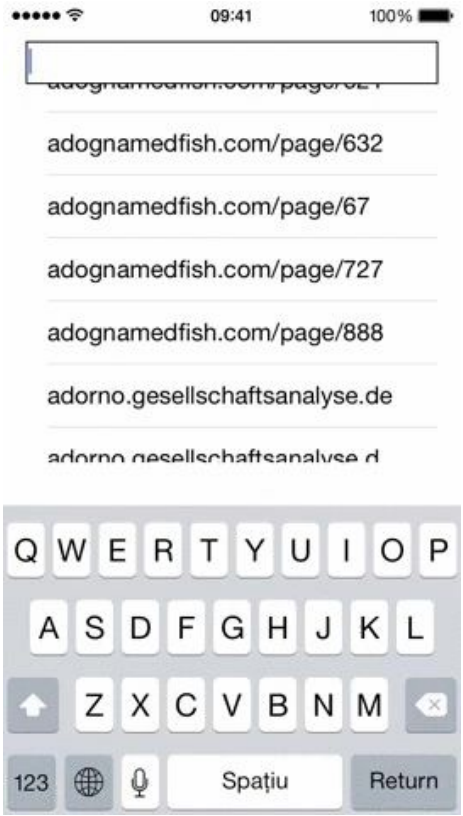
Jeśli chcemy sprawdzić ile prób maksymalnie potrzeba do znalezienia szukanego elementu, to liczymy logarytm o podstawie 2 z liczby n, a następnie (korzystając z wzoru) dodajemy 1 i otrzymujemy wynik.

Jeśli n nie jest potęgą 2, szukamy liczby, którą można przedstawić jako potęgę 2 i która jest mniejsza od n, oraz z wzoru dodajemy 1. Dla otrzymania dokładnego wyniku zaleca się użyć kalkulatora.

Jeśli zbiór ma 256 elementów, to w najgorszym przypadku, ile kroków wyszukiwanie binarne będzie potrzebowało, aby znaleźć określony element?

Obszar zastosowań

- debugowanie kodu
- autouzupełnianie



Przykład w C++ (rozwiązanie iteracyjne)

```
int szukaj(int l, int r, int szukana, int tab[]) // wyszukiwanie binarne (iteracyjnie)
{
  int sr;
  while (l <= r)
  {
        sr = (r + l) / 2; // indeks elementu środkowego przeszukiwanej tablicy
        if (szukana == tab[sr])
              return sr;
        if (szukana < tab[sr])
              r = sr - 1;
        else  l = sr + 1;
  }
  return -1; //jeśli szukanego elementu nie ma w tablicy tp zwróć -1
}
```

http://www.algorytm.edu.pl/algorytmy-maturalne/wyszukiwanie-binarne.html

Przykład w C++ (rozwiązanie rekurencyjne)

```
int szukaj_r(int l, int r, int szukana, int tab[])
{
  if (l > r) // jeśli szukanej liczby nie ma w tablicy to zakończ i zwróć -1
        return -1;

  int sr = (r + l) / 2;

  if (szukana == tab[sr])
        return sr;

  if (szukana < tab[sr])
        szukaj_r(l, sr - 1, szukana, tab); //przeszukuje lewą stronę tablicy
  else
        szukaj_r(sr + 1, r, szukana, tab); //przeszukuje prawą stronę tablicy
}
```

http://www.algorytm.edu.pl/algorytmy-maturalne/wyszukiwanie-binarne.html

https://raw.githubusercontent.com/alicchm892/wdi/master/main.cpp

Dziękujemy za uwagę