

# Fibonačijev hip

Seminarski rad u okviru kursa  
Konstrukcija i analiza algoritama 2  
Matematički fakultet

Ivan Ristović  
Milana Kovacević

januar 2019.

## Sažetak

Fibonačijev hip je struktura podataka osmišljena sa ciljem da poboljša vreme potrebno za operacije nad hipovima. Pružaju bolje amortizovano vreme izvršavanja nego većina drugih prioritetnih redova, uključujući binarni i binomni hip. Fibonačijev hip je osmišljen 1984. godine i publikovan 1987. Ime je dobio po Fibonačijevim brojevima, koji se koriste u analizi složenosti operacija. Koristeći Fibonačijev hip, moguće je unaprediti vremena izvršavanja velikog broja poznatih algoritama kao što je Dijstrin algoritam. Pružamo implementaciju Fibonačijevog hipa u programskom jeziku *Python*, sa interfejsom jednostavnim za upotrebu i testiranje. Takođe u ovom radu testiramo vreme izvršavanja operacije *decrease-key* kako bismo eksperimentalno pokazali konstantno amortizovano vreme izvršavanja ove operacije.

## Sadržaj

<b>1</b>	<b>Uvod</b>	<b>2</b>
<b>2</b>	<b>Opis strukture</b>	<b>2</b>
<b>3</b>	<b>Opis operacija</b>	<b>2</b>
3.1	findmin	2
3.2	extractmin	2
3.3	insert	2
3.4	decreasekey	2
3.5	merge	2
<b>4</b>	<b>Zaključak</b>	<b>3</b>
	<b>Literatura</b>	<b>3</b>

# 1 Uvod

*Binarni hip* (eng. *Heap*) [1] je binarno stablo koje zadovoljava uslov da svaki čvor u stablu ima vrednost ključa veću (tj. manju) od oba svoja sina. Takav hip se često naziva *max-hip* (tj. *min-hip*). Jasno je da će se u hipu maksimum (tj. minimum) alaziti u korenu stabla, što garantuje konstantni upit. Svaki hip podržava sledeće operacije <sup>1</sup>:

- **find\_min** - vraća vrednost ključa korena hipa.
- **extract\_min** - uklanja koren hipa.
- **insert(v)** - unosi novi čvor sa vrednošću ključa  $v$ .
- **decrease\_key(k,v)** - spušta vrednost ključa  $k$  na vrednost  $v$ .
- **merge(h)** - unija sa novim hipom  $h$ .

*Fibonačijev hip* [2] je osmišljen 1984. od strane Fredman-a i Tarjan-a sa ciljem da se poboljša vreme izvršavanja Dijkstrinog algoritma za najkraći put. Originalni Dijkstrin algoritam koji koristi binarni hip radi u vremenskoj složenosti  $O(|E| \log |V|)$ . Korišćenjem Fibonačijevog hipa umesto binarnog hipa, vremensku složenost Dijkstrinog algoritma je moguće poboljšati do  $O(|E| + |V| \log |V|)$ . Poredjenje vremena izvršavanja u odnosu na binarni hip se može videti na sledećoj tabeli:

Operacija	Binarni hip	Fibonačijev hip
<b>find_min</b>	$O(1)$	$O(1)$
<b>extract_min</b>	$O(\log n)$	$O(\log n)$
<b>insert(v)</b>	$O(\log n)$	$O(1)$
<b>decrease_key(k,v)</b>	$O(\log n)$	$O(1)$ <sup>2</sup>
<b>merge(h)</b>	$O(n)$	$O(1)$

**Tabela 1:** Poredjenje vremena izvršavanja operacija izmedju Fibonačijevog i binarnog hipa.

## 2 Opis strukture

## 3 Opis operacija

### 3.1 findmin

### 3.2 extractmin

### 3.3 insert

### 3.4 decreasekey

### 3.5 merge

[3]

---

<sup>1</sup>Pretpostavlja se da se radi o min-hipu, analogno važi i za max-hip.

## 4 Zaključak

### Literatura

- [1] aaa. Heaps. on-line at: <https://www.cs.cmu.edu/~adamchik/15-121/lectures/Binary%20Heaps/heaps.html>.
- [2] Michael L. Fredman, Robert Sedgewick, Daniel D. Sleator, and Robert E. Tarjan. The Pairing Heap: A new form of self-adjusting heap, 1986. on-line at: <http://www.cs.cmu.edu/~sleator/papers/pairing-heaps.pdf>.
- [3] Princeton. Fibonacci Heaps. on-line at: <https://www.cs.princeton.edu/~wayne/teaching/fibonacci-heap.pdf>.