

Heap Sort - sortowanie malejące. Zadanie tablicowe.

Wojciech Szlosek

June 2020

1 Analiza

Przeanalizujemy pewien przykład, analizując go w oparciu o kod programu, który znajduje się jako drugi wysłany plik. help - to moja modyfikacja "downheap" z wykładu. Niech n będzie wielkością kopca (tablicy). Wejściowy przykład: 3 1 9 8 7 2 3 4 5 6

PIERWSZA FAZA:

```
help(4,10): 3 1 9 8 6 2 3 4 5 7
help(3,10): 3 1 9 4 6 2 3 8 5 7
help(2,10): 3 1 2 4 6 9 3 8 5 7
help(1,10): 3 1 2 4 6 9 3 8 5 7
help(0,10): 1 3 2 4 6 9 3 8 5 7
```

Pora na drugą fazę; najpierw "zeswapuj" pierwszy z ostatnim i zmniejsz granicę kopca o 1 ($n-=1$). Następnie przesiej (użyj mojej metody help). Powtarzaj ten krok ($n-2$) razy. Zauważmy, że stopniowo (zaczynając od prawej strony) widzimy już gotowy rezultat sortowania.

DRUGA FAZA:

```
7 3 2 4 6 9 3 8 5 1 (po swapie tab[0] z tab[n-1]; n-=1)
2 3 3 4 6 9 7 8 5 1 (metoda help, "siew")
```

```
5 3 3 4 6 9 7 8 2 1 (po swapie tab[0] z tab[n-1], wcześniej n zostało dekrementowane)
3 4 3 5 6 9 7 8 2 1
```

```
8 4 3 5 6 9 7 3 2 1
3 4 7 5 6 9 8 3 2 1
```

```
8 4 7 5 6 9 3 3 2 1
```

4 5 7 8 6 9 3 3 2 1

9 5 7 8 6 4 3 3 2 1

5 6 7 8 9 4 3 3 2 1

9 6 7 8 5 4 3 3 2 1

6 8 7 9 5 4 3 3 2 1

9 8 7 6 5 4 3 3 2 1

7 8 9 6 5 4 3 3 2 1

9 8 7 6 5 4 3 3 2 1

8 9 7 6 5 4 3 3 2 1

9 8 7 6 5 4 3 3 2 1

9 8 7 6 5 4 3 3 2 1

9 8 7 6 5 4 3 3 2 1

9 8 7 6 5 4 3 3 2 1 (gotowe)

Wszystko działa się następnie analogicznie, więc opisałem tylko dwa pierwsze kroki. Poprzez dekrementację n , stopniowo zbliżamy się do rezultatu, prawa strona tablicy już po kroku pierwszym fazy drugiej była gotowa.