

Drzewa Kopce

Algorytmy i Struktury Danych

Drzewa

– Kopiec (heap)

Kopiec (*ang. heap, tłumaczone też jako stóg lub sterta*)

– struktura danych oparta na drzewie, w której wartości potomków węzła są w **stałej relacji** (**relacja mniejszości, relacja większości**) z wartością rodzica (na przykład wartość rodzica jest nie mniejsza niż wartości jego potomka).

Jeżeli kopiec ma być **kopcem zupełnym**, wtedy dodatkowo spełnione muszą być warunki:

- drzewo jest **prawie pełne**, tzn. liście występują na ostatnim i ewentualnie przedostatnim *poziomie* w drzewie (tylko gdy ostatni poziom nie jest całkowicie wypełniony).
- liście na ostatnim poziomie są spójnie ułożone od strony lewej do prawej

Drzewa

– Kopiec (heap) Właściwości kopców

Jeśli przyjętą relacją między wartością potomka a wartością rodzica będzie **relacja mniejszości**, wówczas na szczycie znajdzie się węzeł z największym kluczem.

Na poziomie i z wyjątkiem ostatniego jest 2^i węzłów.

Wysokość takiego kopca jest logarytmiczna względem jego wielkości $h = \lfloor \log N \rfloor$.

Pełne drzewo binarne o wysokości h składa się z dokładnie $2^{h+1}-1$ elementów.

3

Drzewa

– Kopiec (heap) Właściwości kopców

W przypadku pełnego kopca binarnego, łatwo zaimplementować kopiec w tablicy (o indeksie rozpoczynającym się od 1), według poniższego schematu:

Kolejne elementy numerowane są od wierzchołka, a następnie od lewej do prawej

Następniki węzła k mają numery $2k$ oraz $2k+1$ a poprzednik węzła k ma wartość $k/2$ (dzielenie bez reszty - np. dla 7 rodzic ma numer $7/2 = 3$).

Na każdym kolejnym poziomie kopca możemy łatwo uzyskać dostęp do potomka lewego lub prawego, albo ojca, przy czym:

- Jeśli potomek ma numer n , to ojciec ma $\lfloor n/2 \rfloor$ (np. dla 7 ojciec ma numer 3)
- Jeśli ojciec ma numer n , to lewy potomek ma numer $2n$, a prawy $2n+1$

4

Drzewa

– Kopiec (heap) Właściwości kopców

Wstawianie elementu – *insert*

Wstawiamy element w pierwsze wolne miejsce ostatniego poziomu, a następnie jeżeli wstawiony element jest większy od rodzica, zamieniamy go miejscami. Następnie po zamianie znów sprawdzamy, czy element jest większy od nowego rodzica. Jeżeli tak to znów zamieniamy je miejscami. Operację sprawdzania i zamieniania miejscami powtarzamy tak długo, aż w końcu rodzic będzie większy od wstawionego elementu bądź dojdziemy do korzenia.

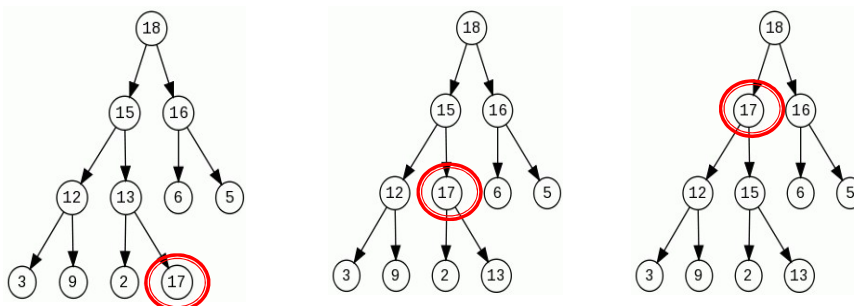
Ponieważ wysokość kopca zupełnego jest logarytmiczna ze względu na ilość węzłów, operacja wstawiania elementu do kopca odbywa się w czasie logarytmicznym.

5

Drzewa

– Kopiec (heap) Właściwości kopców

Wstawianie elementu – *insert*



Wstawiamy element o wartości 17

6

Drzewa

– Kopiec (heap) Właściwości kopców

Usuwanie największego elementu – *delete max*

Usuujemy korzeń. W miejsce korzenia wkładamy prawy skrajny liść. Wymieniamy go miejscami z większym z potomków, następnie z większym z potomków węzła w nowej pozycji.

Operację wykonujemy tak długo aż obaj potomkowie będą mniejsi lub element stanie się liściem.

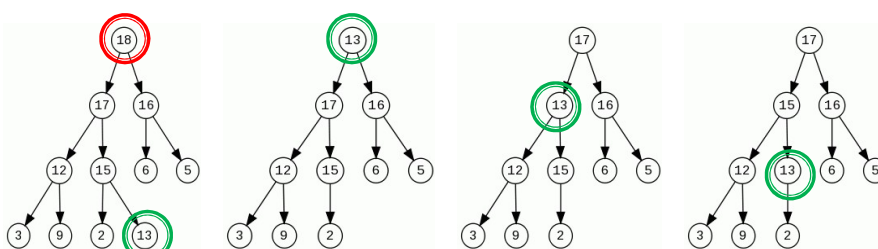
Ponieważ wysokość kopca zupełnego jest logarytmiczna ze względu na ilość węzłów, operacja usuwania największego elementu odbywa się również w czasie logarytmicznym.

7

Drzewa

– Kopiec (heap) Właściwości kopców

Usuwanie największego elementu – *delete max*



8

Drzewa

– Kopiec (heap) Właściwości kopców

Tworzenie kopca – *construct*:

Wykonując ją za pomocą wielokrotnego wstawiania elementów, mielibyśmy czas $O(n \log n)$.

Daje się jednak wykonać kopiec w czasie liniowym działając rekurencyjnie na podkopcach.

Wrzucamy elementy nieuporządkowane. Procedura *construct* wywołuje siebie rekurencyjnie dla prawego i lewego potomka. Wówczas mamy już dwa uporządkowane podkopce. Ich rodzic niekoniecznie jest największy, więc wykonujemy tę samą procedurę przemieszczania go w stronę liścia jak przy *delete max*.

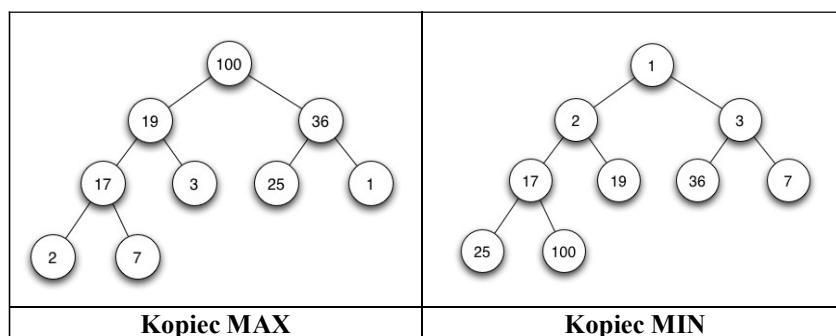
Testy pokazały że naiwna konstrukcja za pomocą wielokrotnego insert jest niewiele wolniejsza w przypadku wstawiania losowych liczb; znacząca różnica staje się przy niekorzystnym przypadku wstawiania kolejnych coraz większych liczb.

9

Drzewa

– Kopiec (heap)

Właściwości kopców



10

DZIĘKUJĘ ZA UWAGĘ