STRUKTURY DANYCH I ZŁOŻONOŚĆ OBLICZENIOWA

Część 7

Kopce

KOPIEC BINARNY(lub STERTA) (*ang.: heap*)

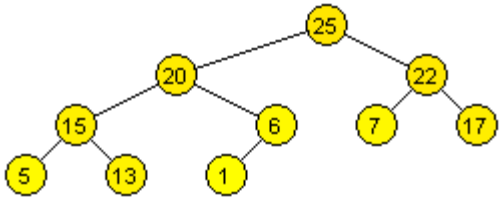
Drzewo binarne jest *kopcem (zorientowanym na maksimum)* wtedy, gdy posiada dwie następujące właściwości:

- **wartość *klucza* przodka jest nie mniejsza niż wartości *kluczy* obu jego potomków;**
- **wszystkie liście kopca leżą co najwyżej na dwóch ostatnich poziomach, a ponadto przedostatni poziom jest całkowicie „wypełniony", zaś na ostatnim poziomie liście „szczelnie" wypełniają jego lewą część.**

Dla kopca nie jest konieczne zachowanie relacji :

klucz lewego potomka ≤ klucz prawego potomka

(z punktu widzenia implementacji bywa nawet „szkodliwe").



Liczba poziomów, na których znajdują się węzły drzewa binarnego o porządku kopcowym, jest najmniejszą z możliwych (a więc kopiec binarny jest *dokładnie wyważony*):

$$h = \lceil lg_2(N+1) \rceil$$

Idea „kopca" jest wykorzystywana wszędzie tam, gdzie z różnych względów istotne jest to, by węzeł o największej (najmniejszej) wartości klucza był korzeniem. Jednym z przykładów jest tzw. *„kolejka priorytetowa"*, w której nie obowiązuje strategia FIFO, lecz wyprowadzanym z kolejki elementem powinien być ten, dla którego wartość klucza jest największa (najmniejsza).

Dygresja

W kopcu mogą istnieć węzły o tej samej wartości klucza (niezależnie od praktycznych konsekwencji tego faktu w rzeczywistych aplikacjach).

Wstawianie nowego węzła do kopca binarnego

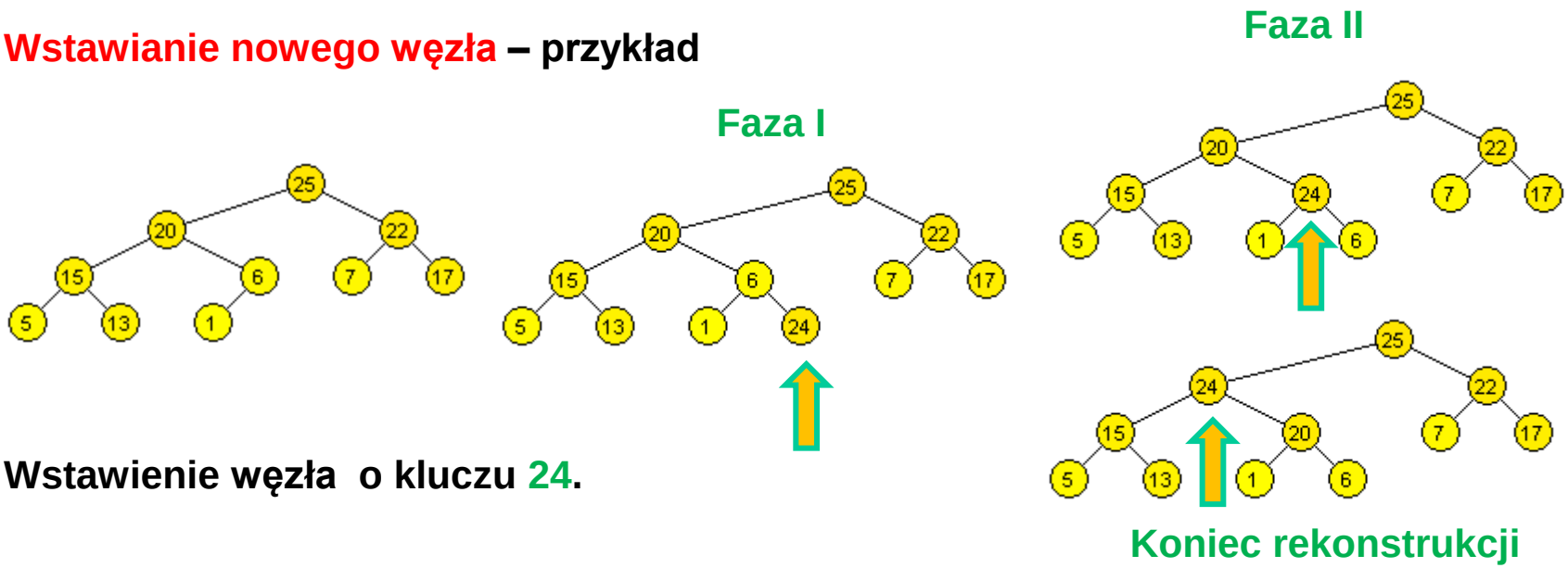
Faza I

Umieszczenie nowego węzła jako kolejnego liścia (licząc od „lewej” strony) na ostatnim poziomie kopca (lub początek obsadzania kolejnego nowego poziomu w kopcu).

Faza II

„Wędrówka” po drodze wiodącej od nowego węzła do korzenia i zamiana miejscami potomka i przodka wtedy, gdy wartość klucza nowego potomka jest większa od wartości klucza przodka (zamiany kończą się po napotkaniu pierwszego przodka o wartości klucza nie mniejszej, niż wartość klucza nowego węzła).

Wstawianie nowego węzła – przykład



Usuwanie węzła z kopca binarnego

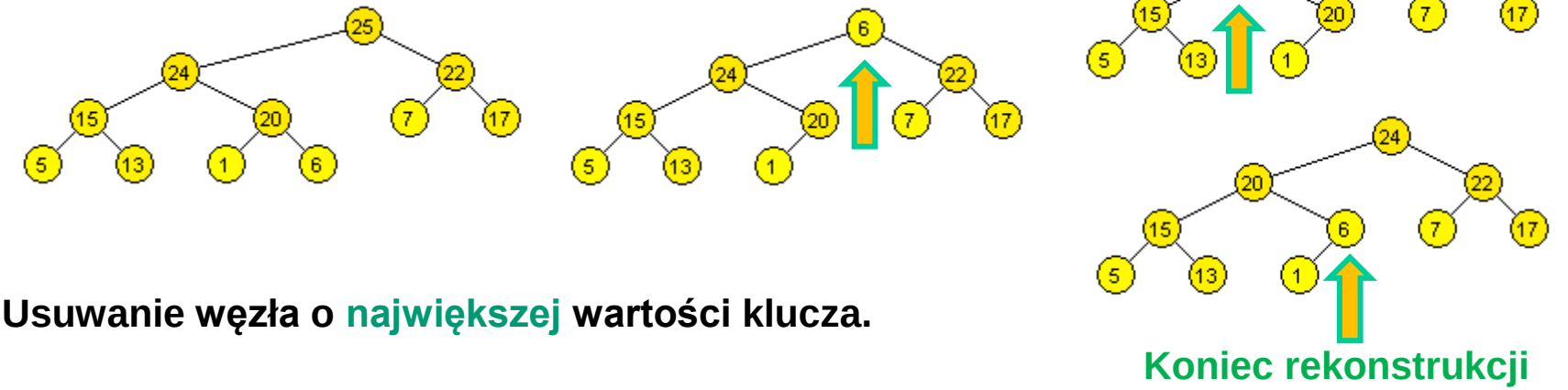
Faza I

Umieszczenie skrajnego prawego węzła z ostatniego poziomu na miejscu węzła usuwanego (zazwyczaj usuwa się z kopca element znajdujący się w korzeniu, chociaż procedura jest identyczna dla dowolnego węzła).

Faza II

„Wędrówka" po drodze wiodącej od przemieszczonego węzła w kierunku potomków i zamiana miejscami potomka oraz przodka wtedy, gdy wartość klucza któregokolwiek z potomków jest większa od wartości klucza przodka (jeżeli oba węzły potomne mają wartości kluczy większe od węzła rodzicielskiego, to wybiera się do zamiany potomka o większej wartości klucza; inny wybór zakłóca porządek kopcowy !!!).

Usuwanie węzła – przykład



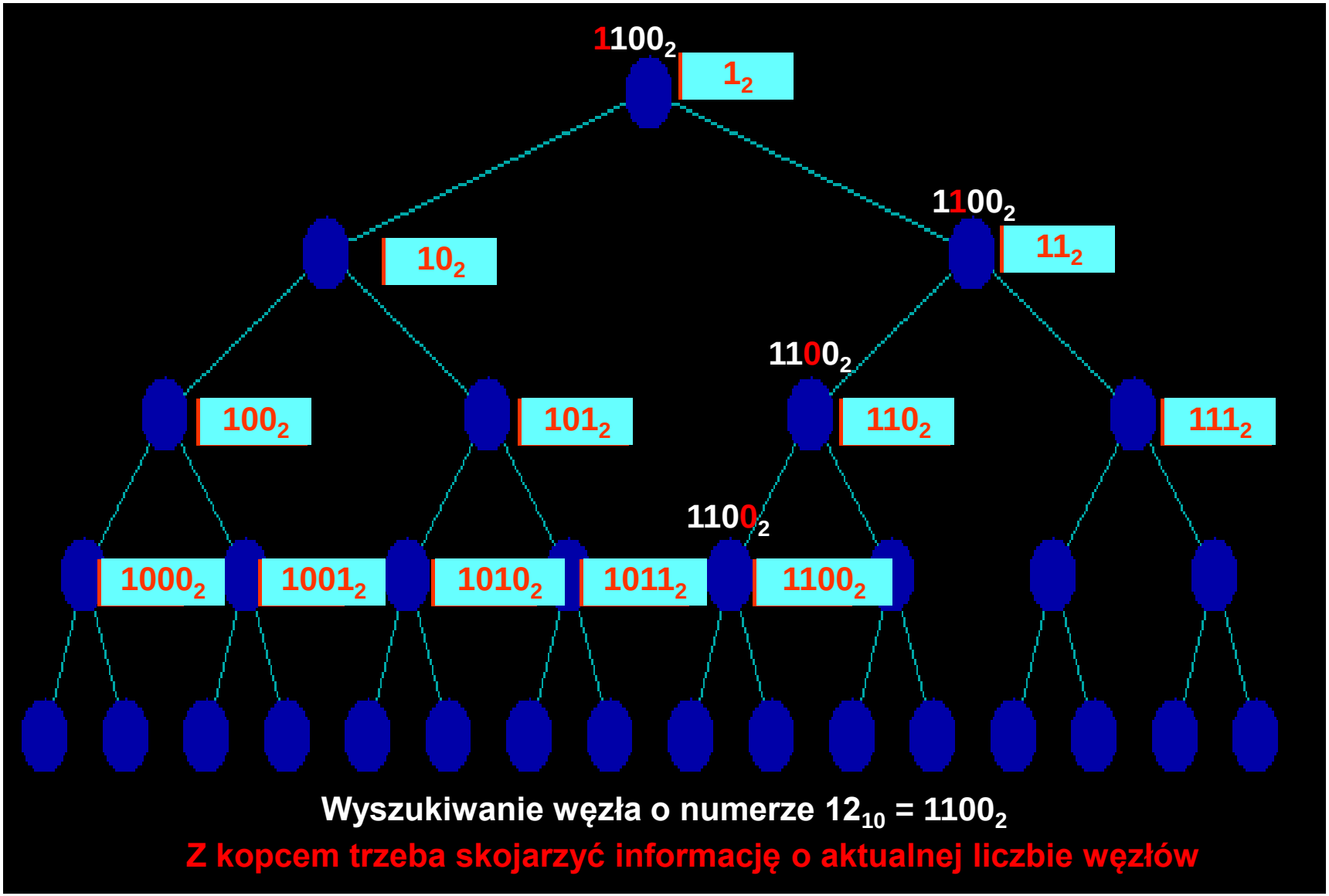
Usuwanie węzła o największej wartości klucza.

Złożoność obliczeniowa operacji z „klasycznym" kopcem binarnym (zorientowanym na maksimum)

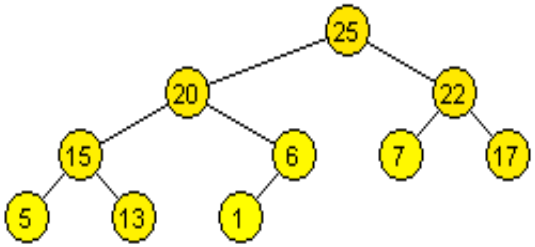
Operacja	*Złożoność (dla N węzłów)*
Wyszukiwanie węzła o największej wartości klucza	*O(1)*
Wyszukiwanie węzła o dowolnej (wskazanej) wartości klucza	*O(N)*
Wstawianie nowego węzła	*O(lg N)*
Usuwanie węzła o największej wartości klucza	*O(lg N)*
Usuwanie węzła o dowolnej (wskazanej) wartości klucza	*O(N)*

Dygresja

Jeżeli jako kryterium poszukiwania przyjąć nie wartość klucza, lecz lokalizację węzła wewnątrz kopca, wówczas operacja wyszukiwania dowolnego węzła w kopcu jest klasy *O(lg N)*.


1100₂
1₂
1100₂
10₂
11₂
1100₂
100₂
101₂
110₂
111₂
1100₂
1000₂
1001₂
1010₂
1011₂
1100₂
Wyszukiwanie węzła o numerze 12₁₀ = 1100₂
Z kopcem trzeba skojarzyć informację o aktualnej liczbie węzłów

Wykorzystanie tablicy w implementacji kopca



Implementacja korzystająca wyłącznie z referencji do kompletnych węzłów (wraz z kluczami, „brelokami” i referencjami do węzłów potomnych).

Jeżeli podczas przywracania porządku kopcowego stosuje się „kopiowanie” składowych, to nie można ograniczyć się do kopiowania kluczy !!!

Implementacja tablicowa – w tablicy przechowywane klucze i „breloki”.

Typ podsta-wowy	klucz + „brelok”	25 + B(25)	20 + B(20)	22 + B(22)	15 + B(15)	6 + B(6)	7 + B(7)	17 + B(17)	5 + B(5)	13 + B(13)	1 + B(1)
Indeks		1	2	3	4	5	6	7	8	9	10

indeks(left_child[i])= 2*indeks(i) **indeks(right_child[i]) = 2*indeks(i)+1**
indeks(parent[i]) = floor(indeks(i)/2)

Implementacja tablicowa – w tablicy przechowywane klucze i referencje do „breloków”.

Typ podsta wowy	klucz + referencja do„breloka”	25 + ref.do B(25)	20 + ref.do B(20)	22 + ref.do B(22)	15 + ref.do B(15)	6 + ref.do B(6)	7 + ref.do B(7)	17 + ref.do B(17)	5 + ref.do B(5)	13 + ref.do B(13)	1 + ref.do B(1)
Indeks		1	2	3	4	5	6	7	8	9	10

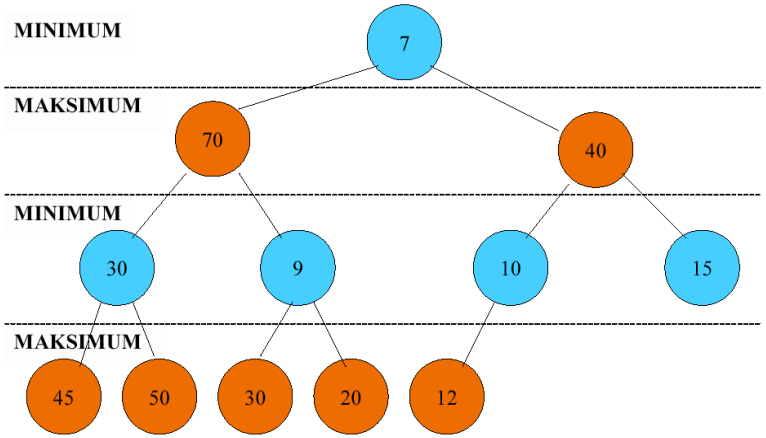
KOPIEC MINIMAKSOWY(*ang.: min-max heap*)

Kopiec minimaksowy ma takie same reguły „obsadzania” węzłami kolejnych poziomów, jak „klasyczny” kopiec binarny.

Poziomy nieparzyste (w tym poziom pierwszy, na którym znajduje się korzeń kopca, o ile kopiec nie jest pusty) noszą nazwę poziomów „minimum”, zaś poziomy parzyste – *poziomów* „maksimum”.

Jeżeli węzeł X jest położony na poziomie „minimum”, to wartość jego klucza jest nie większa od wartości kluczy *we wszystkich węzłach poddrzewa*, dla którego korzeniem jest węzeł X.

Jeżeli węzeł X jest położony na poziomie „maksimum”, to wartość jego klucza jest nie mniejsza od wartości kluczy *we wszystkich węzłach poddrzewa*, dla którego korzeniem jest węzeł X.



Wstawianie nowego węzła do kopca minimaksowego

Faza 1

Umieszczenie nowego węzła jako kolejnego liścia na ostatnim poziomie kopca (analogicznie do „klasycznego” kopca binarnego).

Faza 2

Porównywanie wartości klucza nowo wstawionego węzła z wartościami kluczy „bezpośredniego przodka” i „dziadka”:

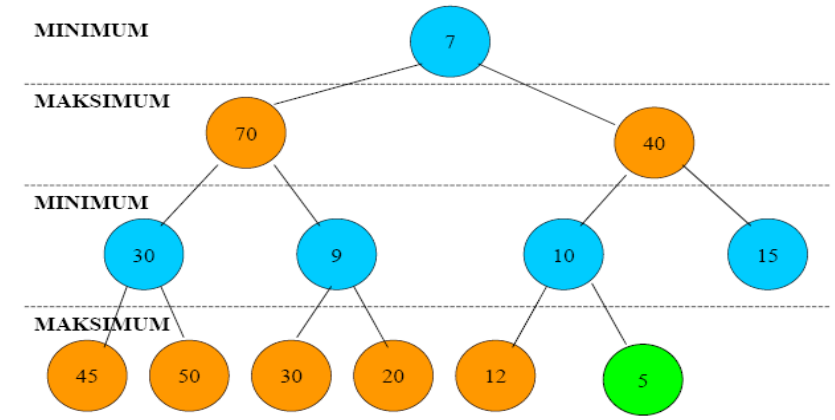
- **jeżeli wartość klucza nowego węzła mieści się w przedziale wyznaczonym przez wartości kluczy obu przodków, to kończy się rekonstrukcję (węzeł pozostaje na swoim miejscu);**
- **jeżeli wartość klucza nowego węzła jest mniejsza od wartości klucza tego z przodków, który znajduje się na poziomie „minimum”, to nowy węzeł jest zamieniany z tym przodkiem miejscami i następuje przejście do fazy 3;**
- **jeżeli wartość klucza nowego węzła jest większa od wartości klucza tego z przodków, który znajduje się na poziomie „maksimum”, to nowy węzeł jest zamieniany z tym przodkiem miejscami i następuje przejście do fazy 3.**

Faza 3

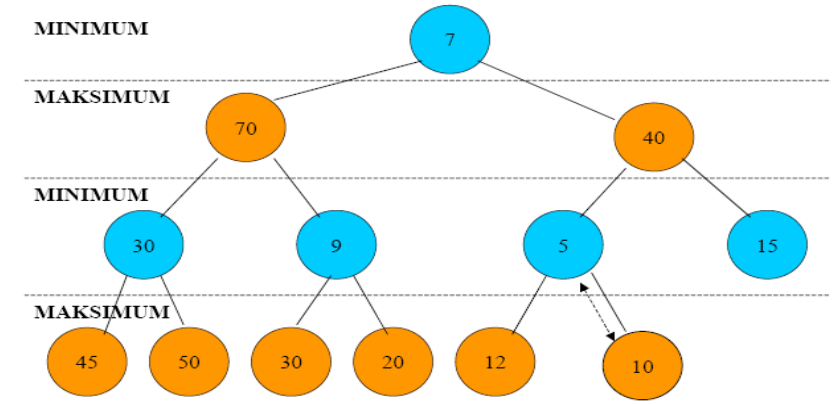
”Wędrówka” w górę kopca i kolejne zamiany (dopóki nie zostanie przywrócony porządek kopcowy lub nie dotrze się do korzenia, albo jego bezpośrednich potomków znajdujących się na poziomie „maksimum”); w tej fazie analizowane są wyłącznie relacje kluczy „wędrującego” węzła i kluczy węzłów na poziomach takich samych, jak zajęty przez nowy węzeł po zakończeniu fazy 2 („maksimum” albo „minimum” – porównania dokonywane są co drugi poziom).

Wstawianie nowego węzła do kopca minimaksowego – przykład 1

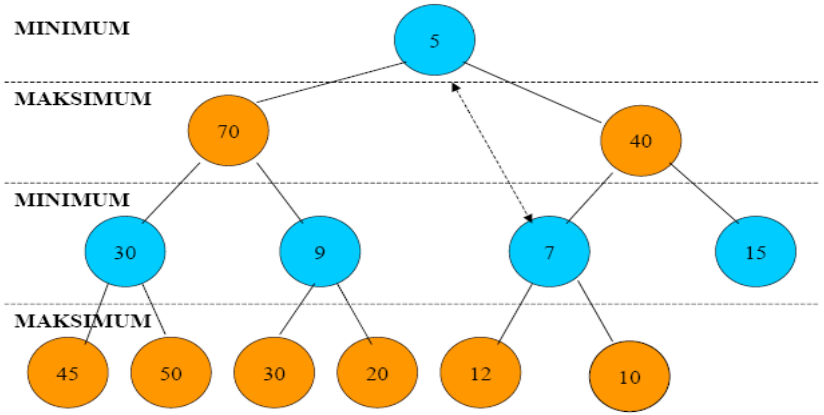
faza 1



faza 2

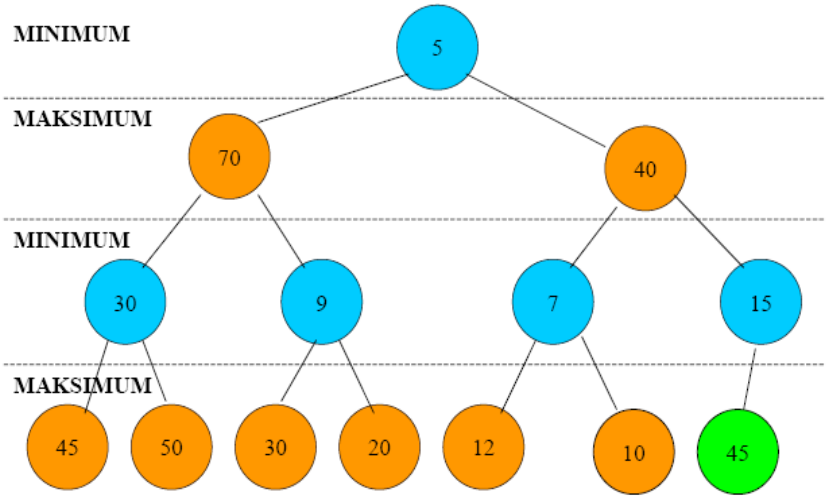


faza 3(i koniec rekonstreukcji)

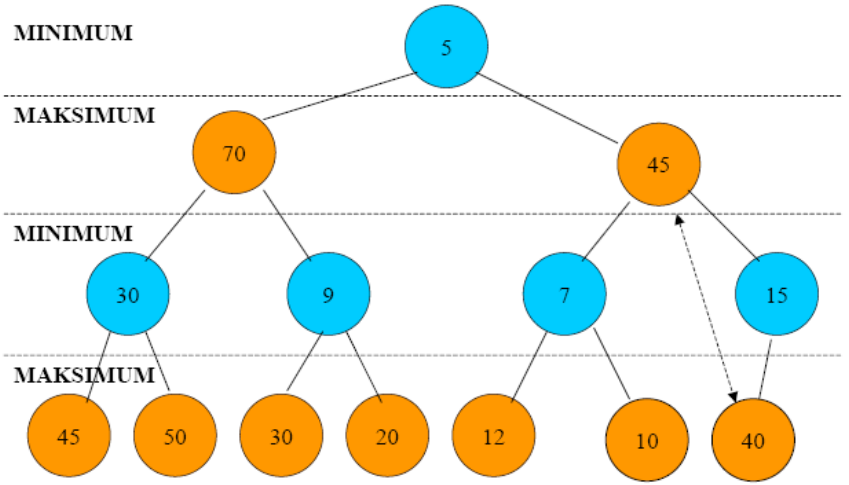


Wstawianie nowego węzła do kopca minimaksowego – przykład 2

faza 1



faza 2 (i koniec rekonstrukcji – brak fazy 3)



Usuwanie węzła o najmniejszej (największej) wartości klucza z kopca minimaksowego

Faza 1

Usunięcie węzła (węzeł o najmniejszej wartości klucza jest korzeniem; węzeł o największej wartości klucza jest jednym z dwóch jego potomków – należy wybrać jednego z nich porównując ich klucze) i umieszczenie na jego miejscu skrajnego prawego liścia z ostatniego poziomu kopca (niezależnie od poziomu, na którym się znajduje).

Faza 2

”Wędrówka” w dół kopca i kolejne zamiany (dopóki nie osiągnie się porządku kopcowego lub nie dotrze się do poziomu *ostatniego* lub *przedostatniego*);

- **jeżeli osiągnięto jeden z dwóch ostatnich poziomów, to następuje przejście do fazy 3;**
- **jeżeli przed osiągnięciem poziomu ostatniego lub przedostatniego przywrócono porządek kopcowy, to następuje *koniec rekonstrukcji*.**

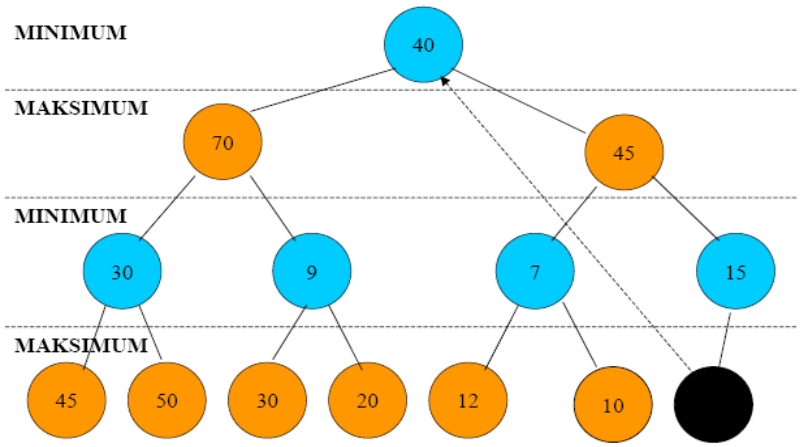
W tej fazie analizowane są *wyłącznie relacje kluczy „wędrującego” węzła i kluczy węzłów na poziomach takich samych, jak zajęty przez węzeł „wypromowany” na poziom pierwszy lub drugi w fazie 1* („maksimum” albo „minimum” – porównania dokonywane są co drugi poziom);

Faza 3

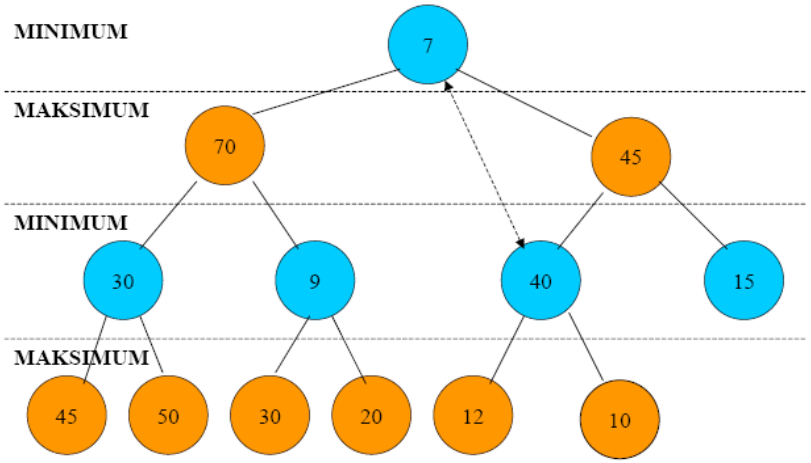
Porównanie wartości klucza „wędrującego” węzła z kluczami bezpośredniego przodka i ewentualnych potomków; jeżeli jest to konieczne dla zachowania porządku w kopcu minimaksowym, to dokonuje się zamiany węzłów.

Usuwanie węzła o najmniejszej wartości klucza – przykład

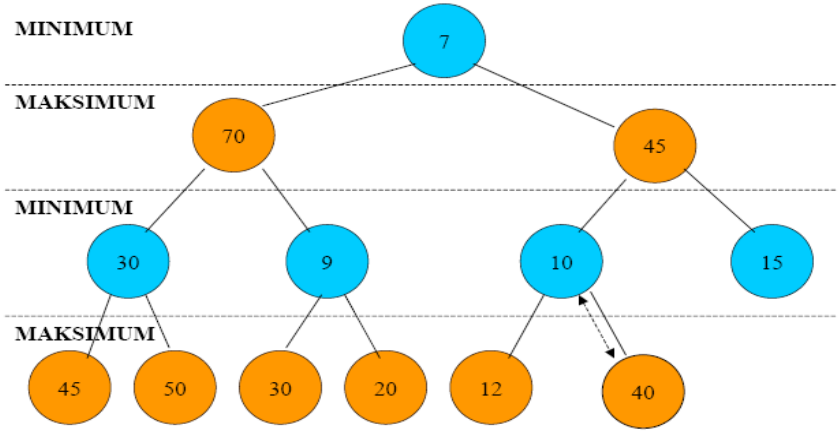
faza 1



faza 2

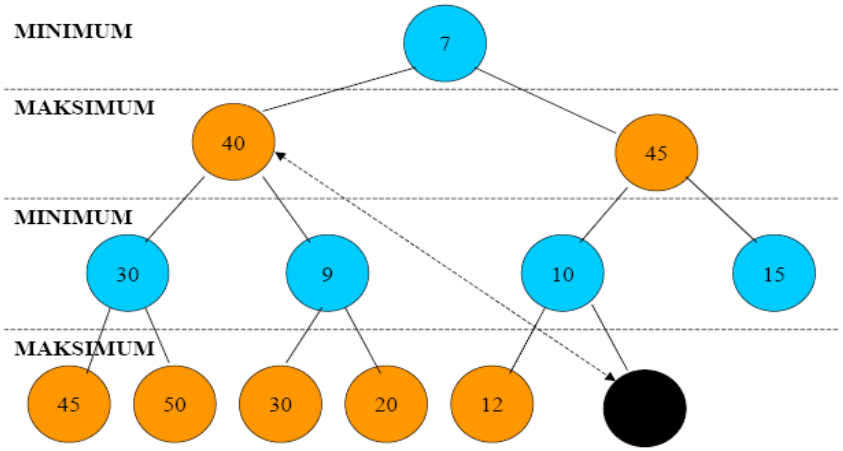


faza 3 (i koniec rekonstrukcji)

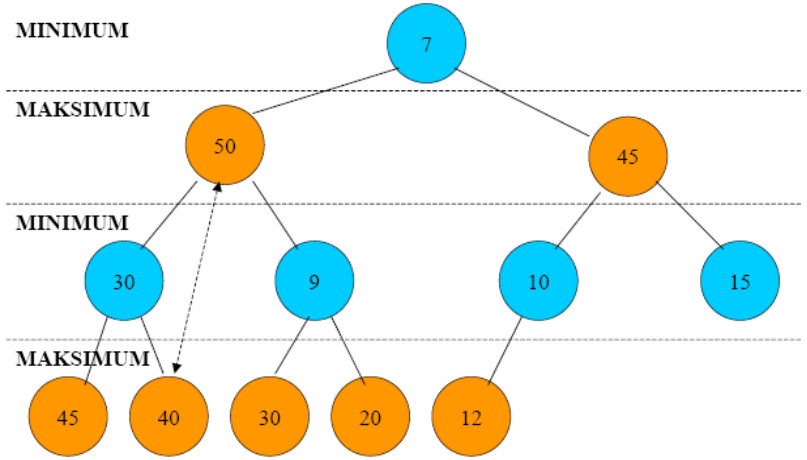


Usuwanie węzła o największej wartości klucza – przykład

faza 1



faza 2 (i koniec rekonstrukcji – w fazie 3 brak zamian)



KOPIEC SPRZĘŻONY *(ang.: deap heap)*

Kopiec sprzężony pod względem „obsadzenia” kolejnych poziomów przez węzły jest także identyczny, jak „klasyczny” kopiec binarny.

Korzeń kopca *nie zawiera żadnego elementu danych* (ale istnieje !!!).

Lewe poddrzewo korzenia jest „klasycznym” *kopcem binarnym zorientowanym na minimum*, zaś prawe poddrzewo – „klasycznym” *kopcem binarnym zorientowanym na maksimum*.

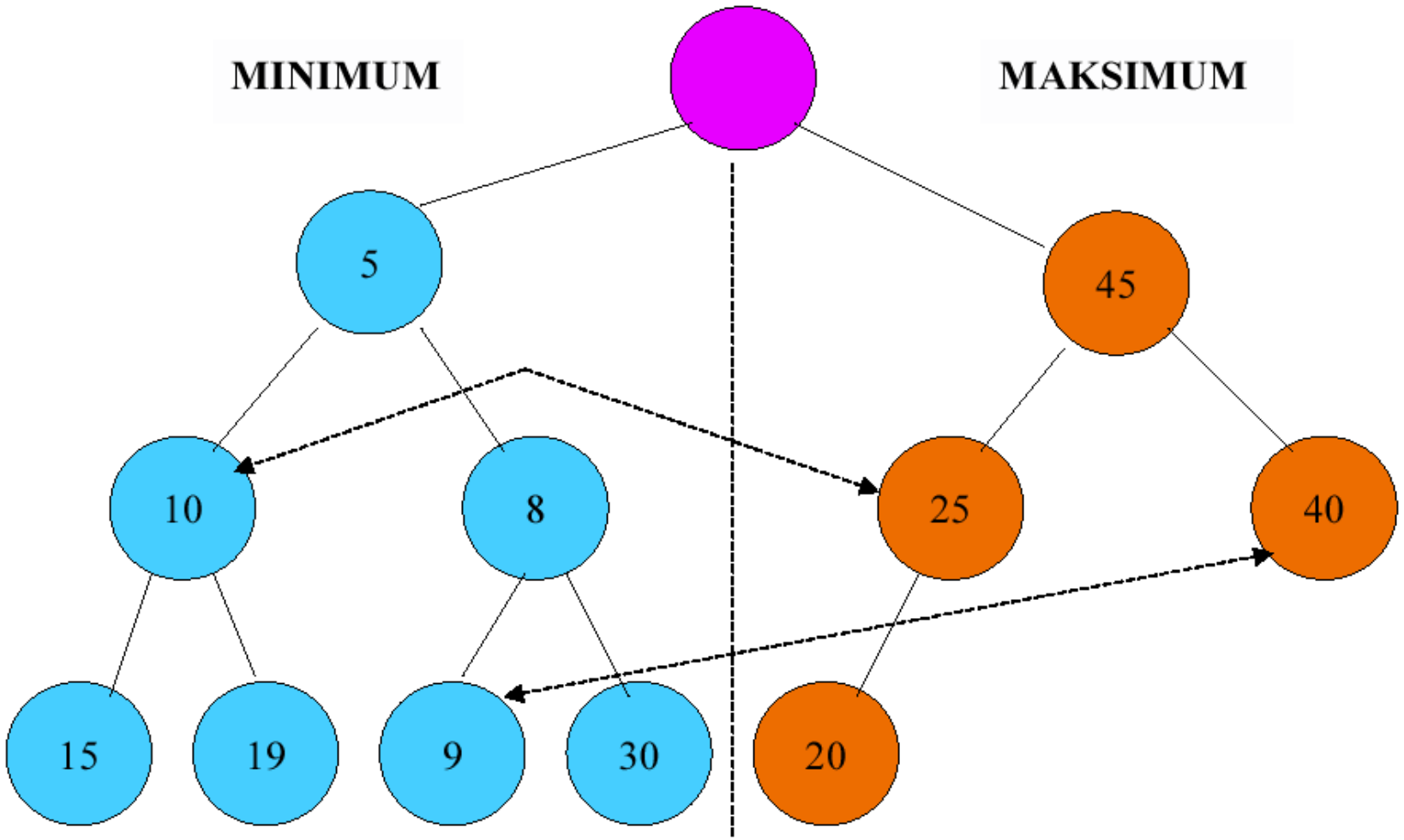
Każdy z węzłów prawego poddrzewa korzenia (kopca „maksimum”) ma swojego *„kuzyna”* w lewym poddrzewie korzenia (kopcu „minimum”).

„Kuzyn” jest węzłem zajmującym „*topologicznie*” to samo miejsce w „lustrzanym” kopcu.

Węzły kopca „minimum” nie muszą mieć tak określonych „kuzynów”, ale jeżeli ich brak, to rolę „kuzynów” przejmują *potencjalni przodkowie tych „kuzynów”*.

Wartość klucza tego z „kuzynów”, który znajduje się w kopcu „minimum”, jest nie większa od wartości klucza „kuzyna” z kopca „maksimum”.

Przykład (przerywanymi strzałkami połączono niektóre pary „kuzynów”):



Wstawianie nowego węzła do kopca sprzężonego

Faza 1

Umieszczenie nowego węzła jako kolejnego liścia na ostatnim poziomie kopca (analogicznie do „klasycznego” kopca binarnego).

Faza 2

Porównanie wartości klucza nowo wstawionego węzła z wartością klucza „kuzyna”:

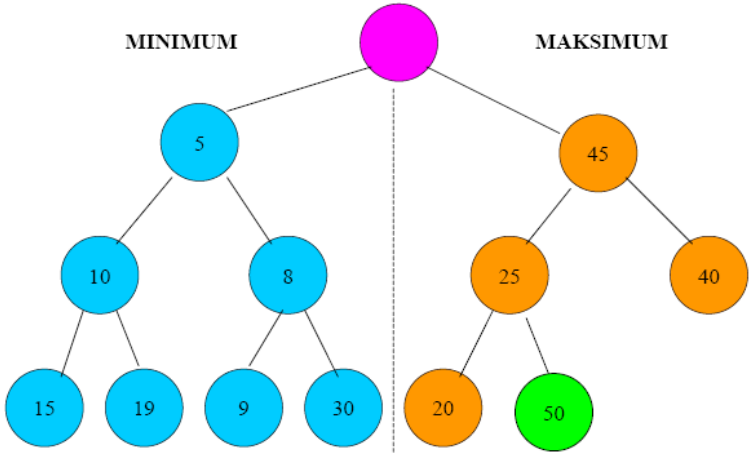
- **jeżeli nowy węzeł został umieszczony w kopcu „minimum”, zaś wartość jego klucza jest nie większa od wartości klucza „kuzyna” i nie mniejsza od wartości klucza bezpośredniego przodka (i analogicznie: jeżeli nowy węzeł został umieszczony w kopcu „maksimum”, zaś wartość jego klucza jest nie mniejsza od wartości klucza „kuzyna” i nie większa od wartości klucza bezpośredniego przodka), to kończy się rekonstrukcję (węzeł pozostaje na swoim miejscu);**
- **jeżeli nowy węzeł został umieszczony w kopcu „minimum”, zaś wartość jego klucza jest większa od wartości klucza „kuzyna”, to nowy węzeł jest zamieniany z „kuzynem” (zajmuje odpowiednie miejsce w kopcu „maksimum”) i następuje przejście do fazy 3;**
- **jeżeli nowy węzeł został umieszczony w kopcu „maksimum”, zaś wartość jego klucza jest mniejsza od wartości klucza „kuzyna”, to nowy węzeł jest zamieniany z „kuzynem” (zajmuje odpowiednie miejsce w kopcu „minimum”) i następuje przejście do fazy 3.**

Faza 3

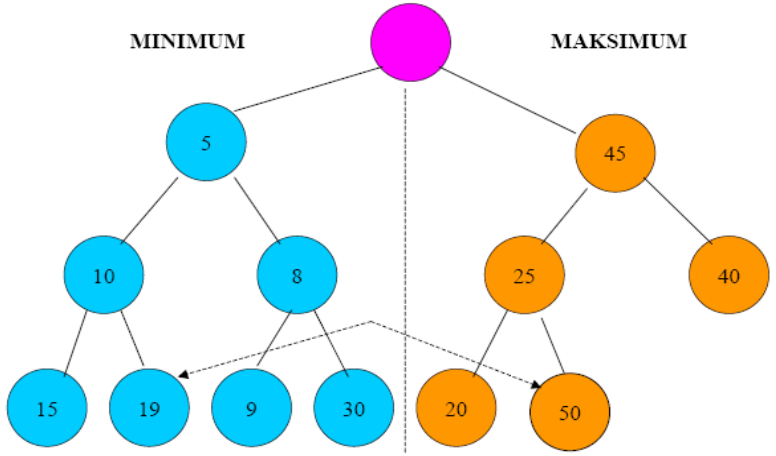
”Wędrówka” nowo wstawionego węzła w górę kopca i kolejne zamiany (dopóki nie zostanie przywrócony porządek kopcowy lub nie dotrze się do lewego albo prawego potomka korzenia).

Wstawianie nowego węzła do kopca sprzężonego – przykład 1

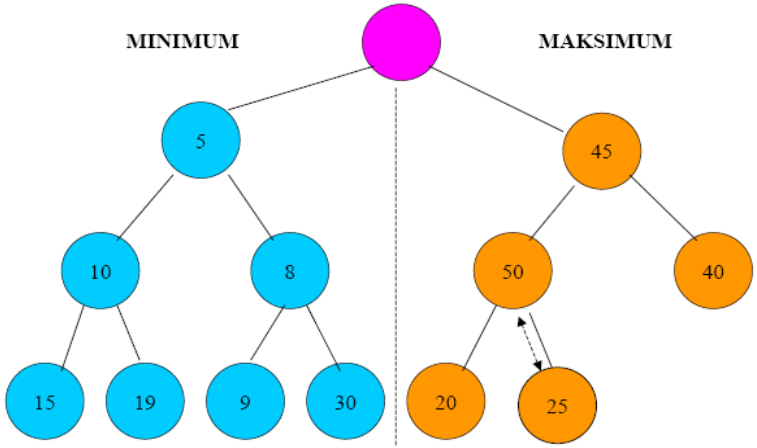
faza 1



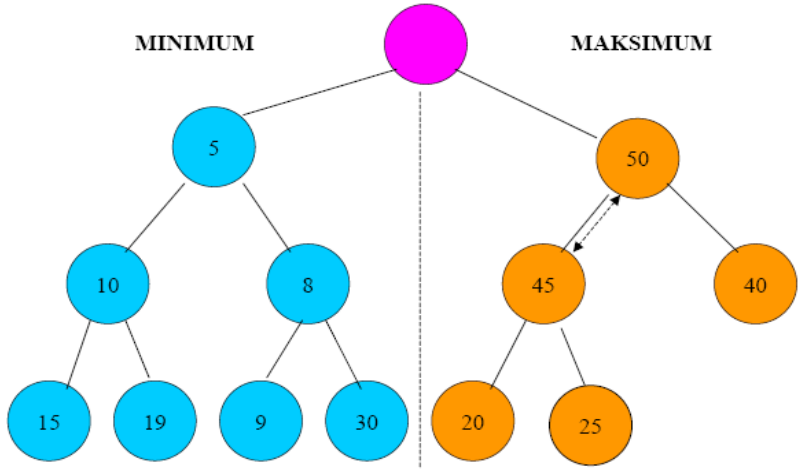
faza 2 (brak zamiany z „kuzynem")



faza 3 (pierwsza zamiana w „wędrówce" ku korzeniowi)

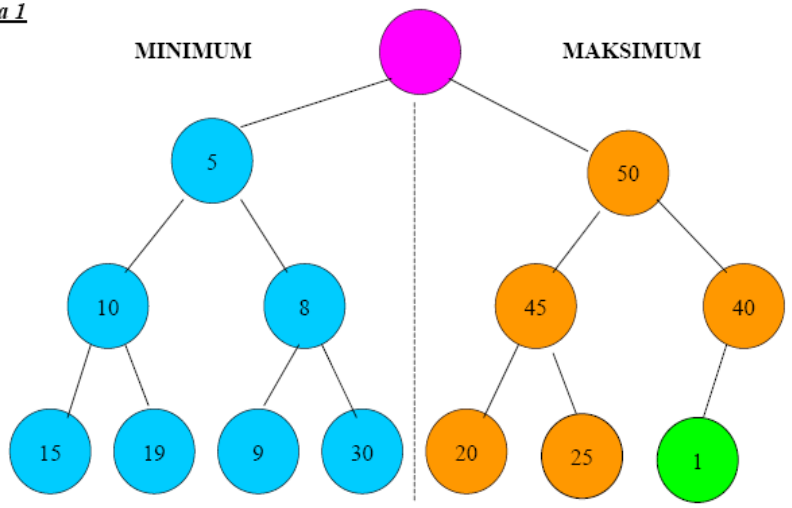


faza 3 (druga zamiana w „wędrówce" ku korzeniowi i koniec rekonstrukcji)

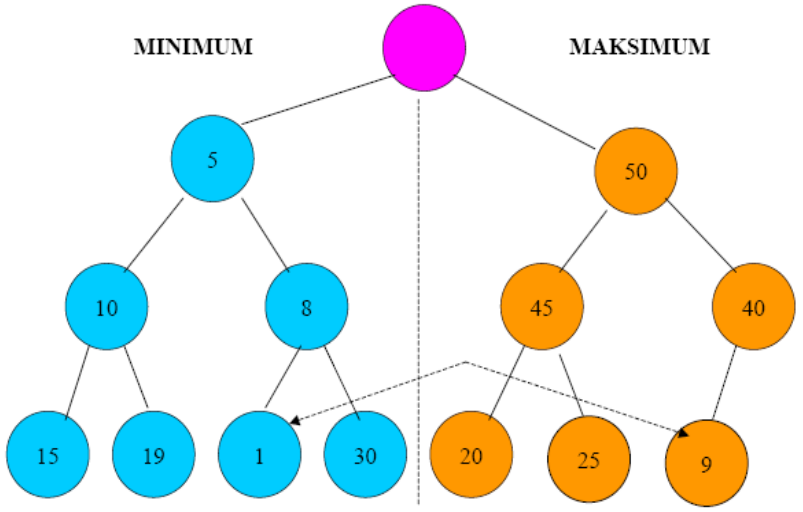


Wstawianie nowego węzła do kopca sprzężonego – przykład 2

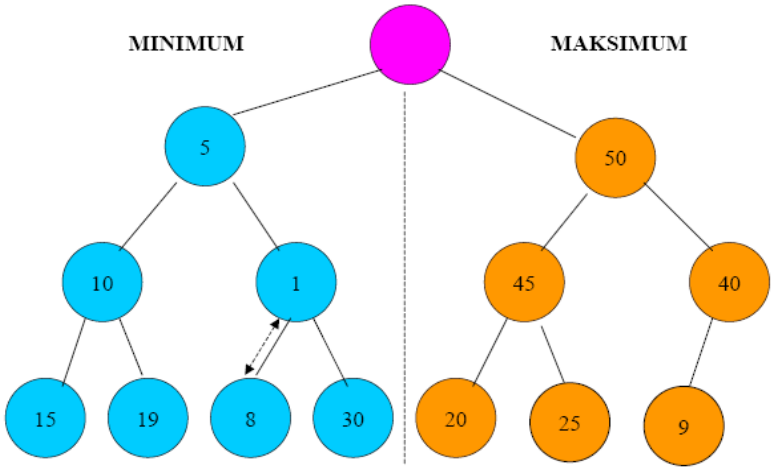
faza 1



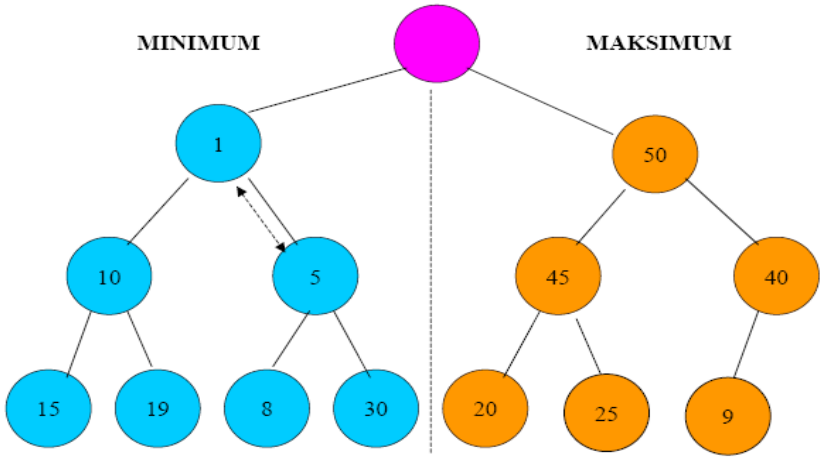
faza 2 (zamiana z „kuzynem")



faza 3 (pierwsza zamiana w „wędrówce" ku korzeniowi)



faza 3 (druga zamiana w „wędrówce" ku korzeniowi i koniec rekonstrukcji)



Usuwanie węzła o najmniejszej (największej) wartości klucza z kopca sprzężonego

Faza 1

Usunięcie węzła (węzeł o najmniejszej wartości klucza jest lewym potomkiem korzenia; węzeł o największej wartości klucza jest prawym potomkiem korzenia) i umieszczenie na jego miejscu skrajnego prawego liścia z ostatniego poziomu całego kopca (niezależnie od tego, czy znajduje się w kopcu „minimum”, czy w kopcu „maksimum”).

W przypadku, gdy kopiec zawiera tylko jeden węzeł, to usuwa się go i kończy rekonstrukcję.

Faza 2

”Wędrówka” w dół kopca i kolejne zamiany (dopóki nie osiągnie się porządku kopcowego lub nie dotrze się do ostatniego poziomu);

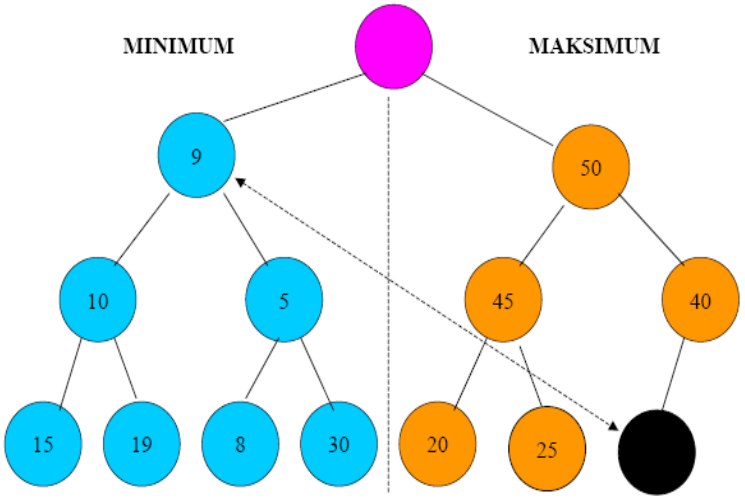
- **jeżeli osiągnięto ostatni poziom, to następuje przejście do fazy 3;**
- **jeżeli przed osiągnięciem ostatniego poziomu przywrócono porządek kopcowy, to następuje koniec rekonstrukcji.**

Faza 3

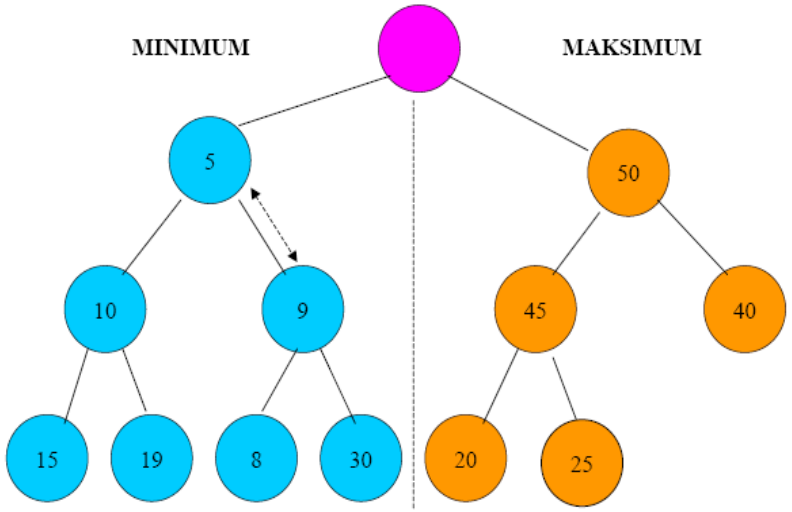
Porównanie wartości klucza „wędrującego” węzła z kluczem „kuzyna”; jeżeli jest to konieczne dla zachowania porządku w kopcu sprzężonym, to dokonuje się zamiany „kuzynów”.

Usuwanie węzła o najmniejszej wartości klucza – przykład

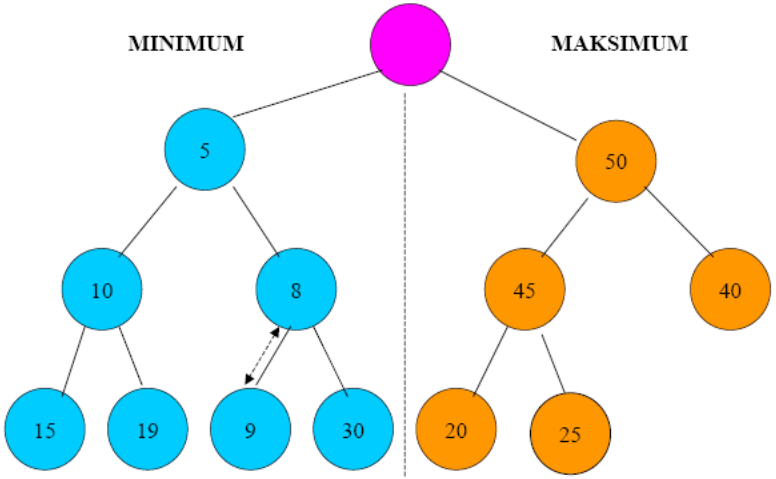
faza 1



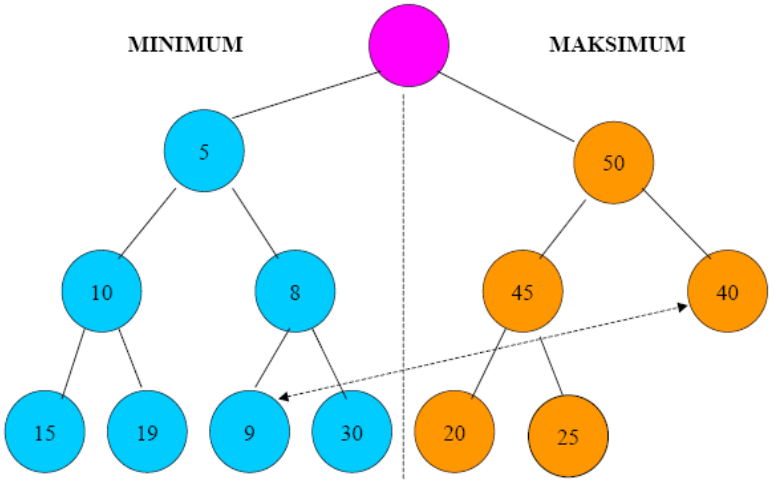
faza 2 (pierwsza zamiana w „wędrówce" ku poziomowi liści)



faza 2 (druga zamiana w „wędrówce" ku poziomowi liści – koniec fazy 2.)

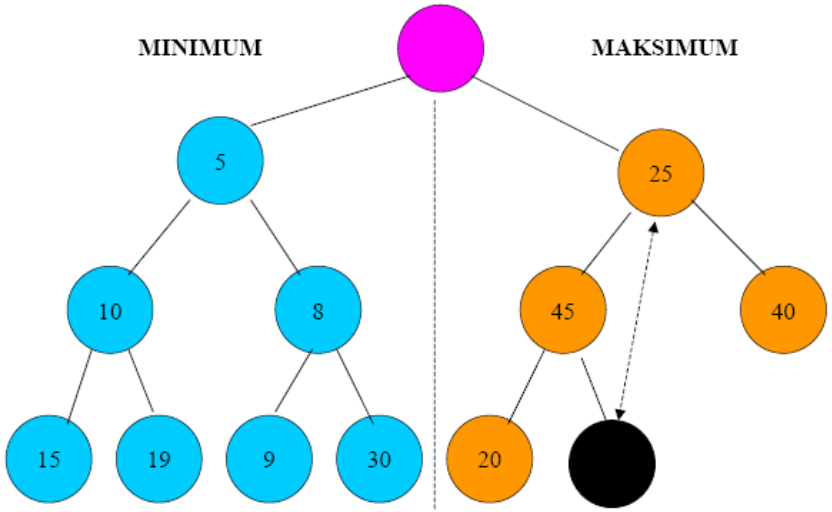


faza 3 (porównanie kluczy kuzynów; brak zamiany i koniec rekonstrukcji)

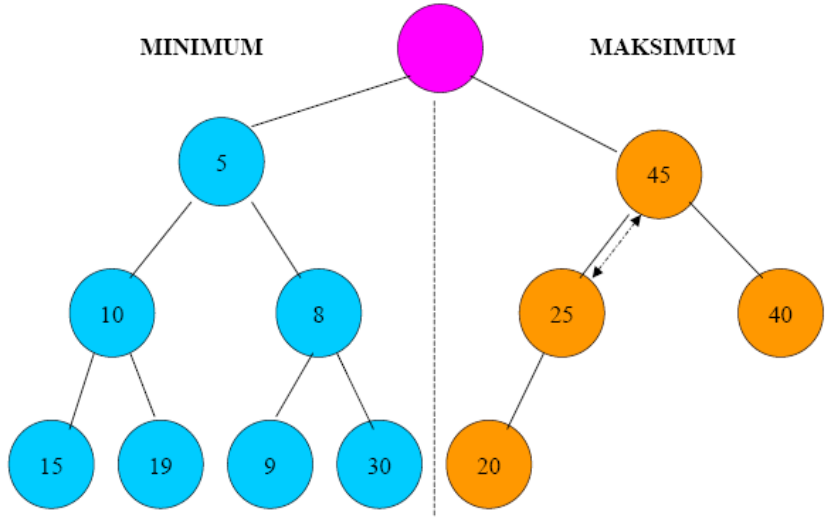


Usuwanie węzła o największej wartości klucza – przykład

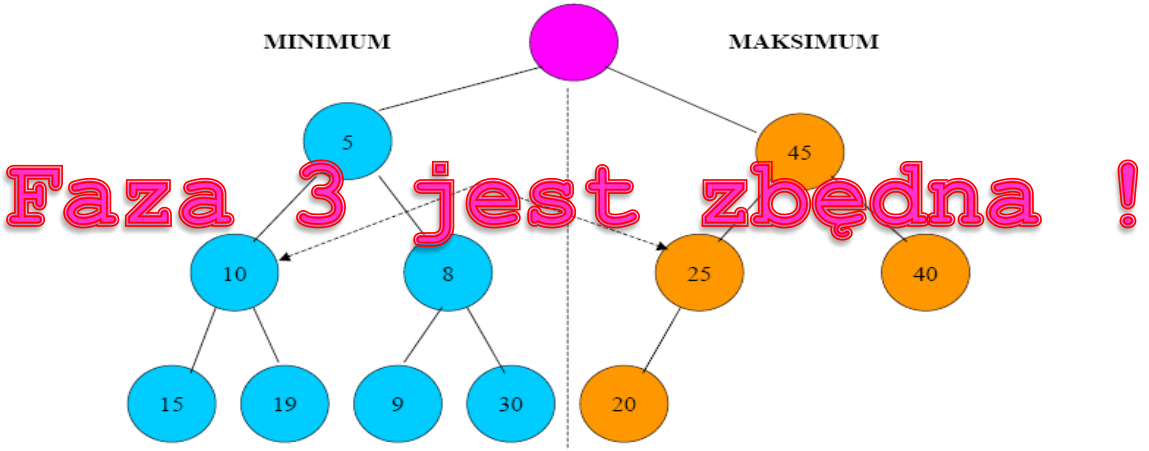
faza 1



faza 2 (zamiana i koniec fazy 2.)



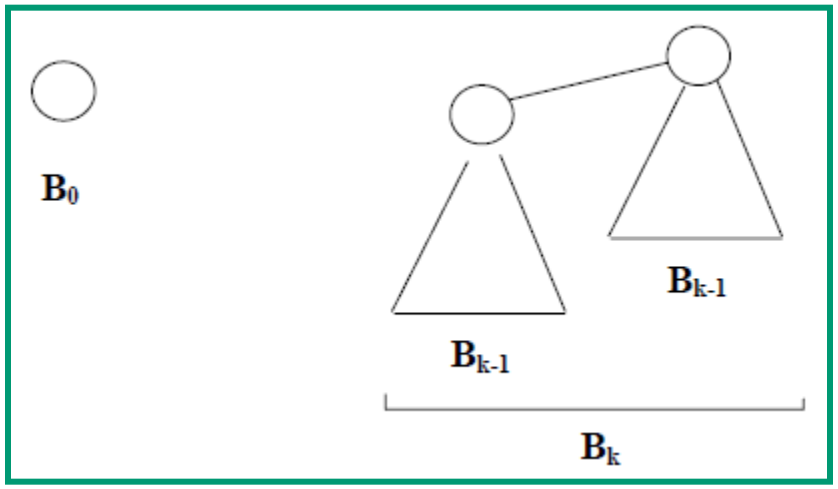
faza 3 (porównanie kluczy kuzynów; brak zamiany i koniec rekonstrukcji)

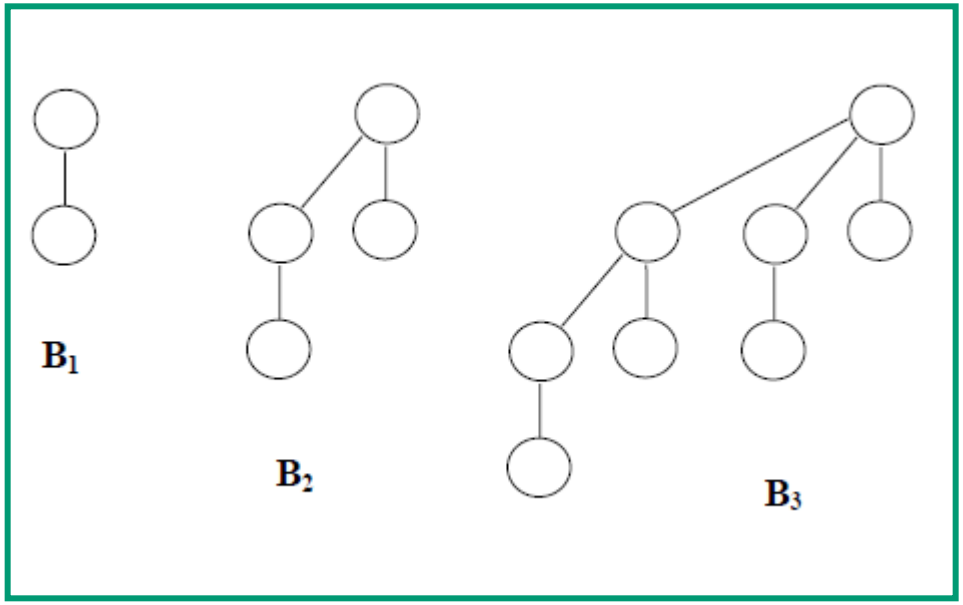


DRZEWO DWUMIANOWE (*ang.: binomial tree*)

Drzewo dwumianowe B_k jest drzewem zdefiniowanym rekurencyjnie w sposób następujący:

- **drzewo dwumianowe B_0 składa się z jednego węzła (korzenia);**
- **drzewo dwumianowe B_k składa się z dwóch drzew dwumianowych B_{k-1} powiązanych tak, że korzeń jednego z nich jest lewym skrajnym potomkiem drugiego.**



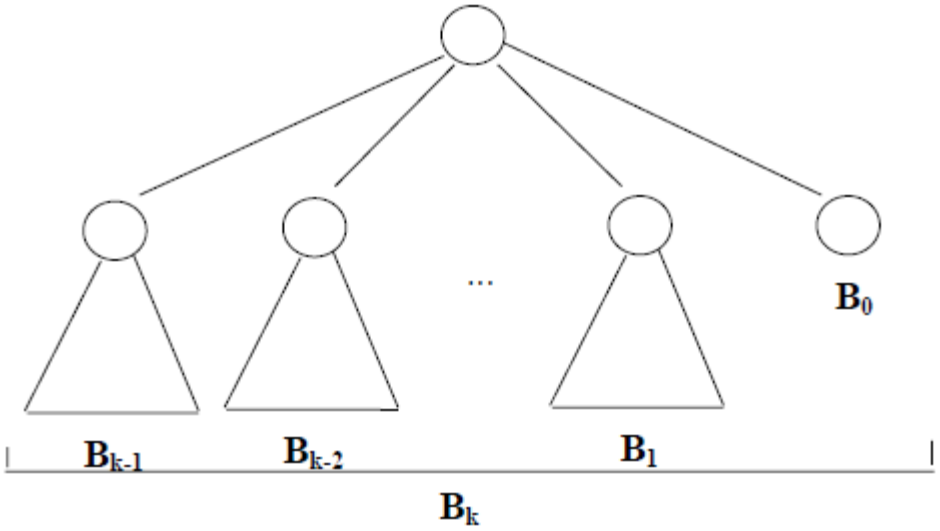


Własności drzew dwumianowych

- drzewo dwumianowe B_k posiada dokładnie 2^k węzłów;
- wysokość drzewa B_k wynosi $(k + 1)$;
- na poziomie i-tym znajduje się dokładnie $\binom{k}{i-1}$ węzłów , (stąd min. wywodzi się nazwa drzewa);
- stopień korzenia wynosi k, zaś wszystkie pozostałe węzły, jeżeli istnieją, mają stopień co najwyżej $(k – 1)$;
- maksymalny stopień węzła w drzewie dwumianowym zawierającym n węzłów wynosi $\lg_2 n$;
- jeżeli ponumerować potomków korzenia kolejno:

  $k - 1, k - 2, ..., 2, 1, 0,$

  to potomek o numerze i-tym jest korzeniem drzewa dwumianowego B_i.

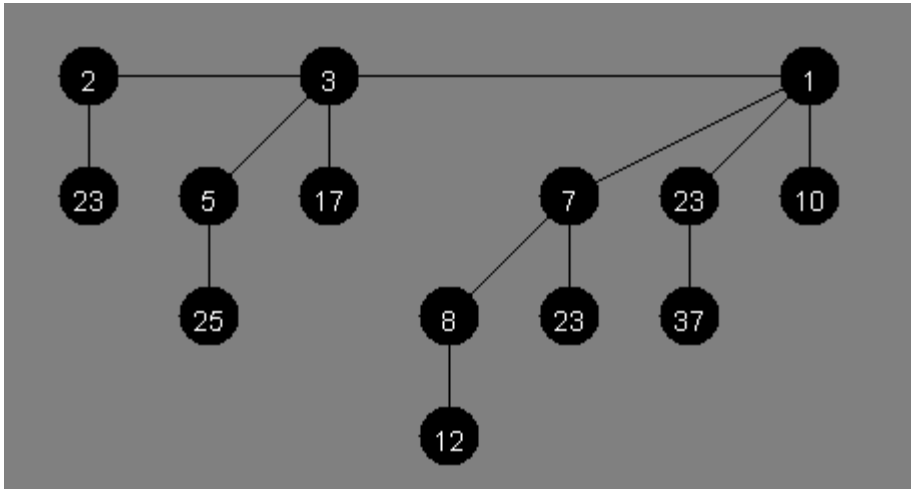


KOPIEC DWUMIANOWY (*ang.: binomial heap*)

Kopiec dwumianowy H jest zbiorem drzew dwumianowych o następujących właściwościach:

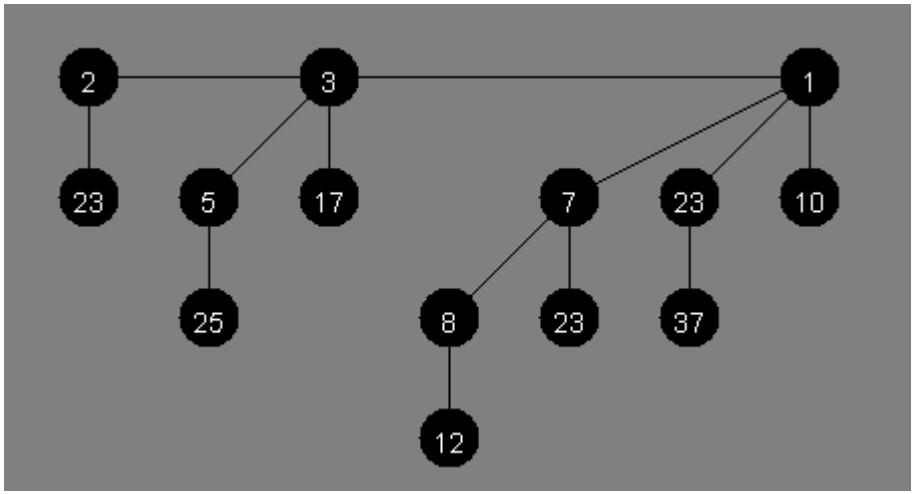
- **każde drzewo dwumianowe jest uporządkowane kopcowo, tzn. klucz w węźle jest nie mniejszy (lub w wersji alternatywnej: nie większy) niż klucz przodka;**
- **dla każdego $d \geq 0$ istnieje w kopcu H co najwyżej jedno drzewo dwumianowe B_d;**
- **korzenie drzew dwumianowych tworzących kopiec H są uporządkowane w formie listy, przy czym ich kolejność jest zgodna ze wzrostem stopnia korzeni poszczególnych drzew składowych.**

Przykład kopca dwumianowego



Dygresja

Jeżeli obecność drzew B_i w kopcu reprezentować w postaci liczby binarnej, w której wagi poszczególnych pozycji odpowiadają indeksowi drzewa, to liczba wszystkich węzłów w kopcu jest jednoznacznie określona przez tą liczbę.



W powyższym przykładzie: $(1110)_2 = (8+4+2)_{10} = (14)_{10}$

Należy także zwrócić uwagę na fakt, że wartości kluczy potomków należących do tego samego poziomu drzewa dwumianowego (czyli „braci") nie mają wpływu na ich uporządkowanie.

To samo dotyczy kluczy korzeni drzew dwumianowych tworzących kopiec.

Reprezentacja węzłów w kopcach dwumianowych musi uwzględniać wskaźniki na przodka i „prawego brata”.

Propozycja struktury dla elementu typu podstawowego :

```
struct node_rec
        {
        int index;//stopień węzła, informuje o liczbie potomków
        eltype key;
        struct node_rec *parent;
        struct node_rec *brother;      //wskaźnik na prawego brata
        struct node_rec *left_child;   //wskaźnik na „pierwszego”
                                       //lewego potomka
        struct data_rec;               //dane nie wpływające
                                       //na uporządkowanie drzewa
        };
```

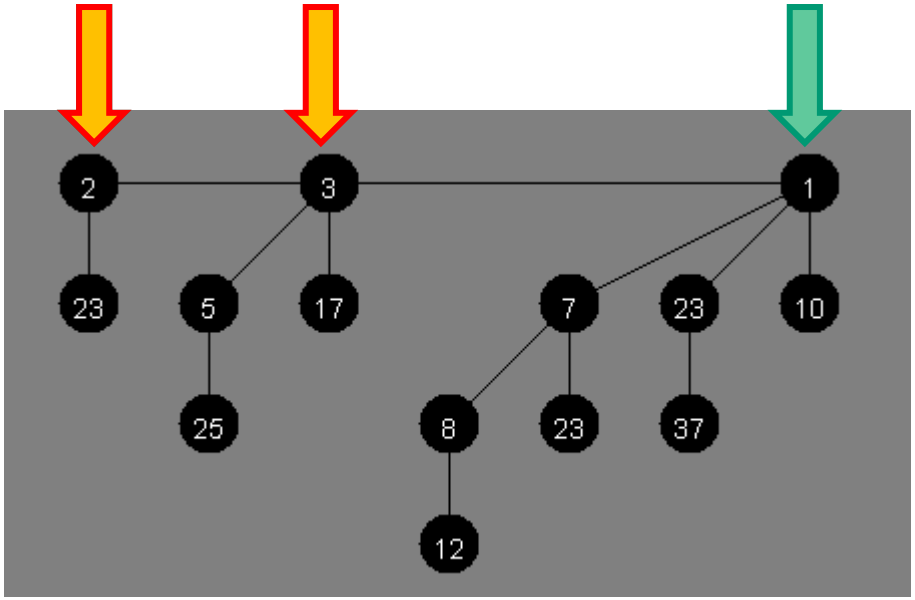
Wskaźnik na prawego brata wskazuje na:

- **korzeń kolejnego drzewa dwumianowego w liście (dla korzenia);**
- **kolejny węzeł potomny dla tego samego przodka (potomkowie nie musza być liniowo uporządkowani, wystarczy uporządkowanie kopcowe);**
- **NULL, gdy nie istnieje węzeł spełniający żadnego z powyższych dwóch warunków.**

Znajdowanie węzła o najmniejszej wartości klucza

Dla każdego z drzew dwumianowych tworzących kopiec element o najmniejszej wartości klucza znajduje się w korzeniu, zatem w celu znalezienia takiego węzła wystarczy poruszać się wzdłuż listy, na której znajdują się korzenie (czyli po ścieżce wskazywanej przez wskaźniki na kolejnych prawych braci korzeni).

Przykład



Złożoność $O(\lg N)$

Łączenie dwóch kopców dwumianowych

Operacja łączenia dwóch kopców dwumianowych składa się z dwóch faz:

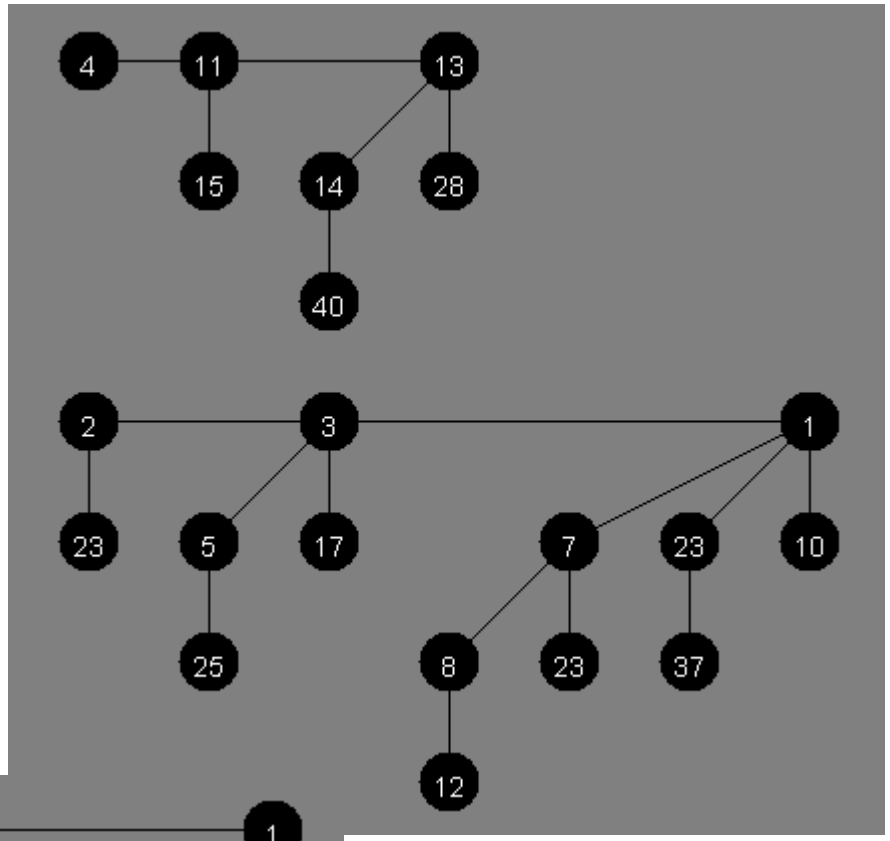
- łączenia kopców tak, by korzenie drzew dwumianowych obu kopców tworzyły listę, na której stopnie korzeni uporządkowane są w porządku narastającym (z reguły zakłada się, że jeżeli w obu kopcach występują dwa drzewa o tym samym stopniu korzenia, to korzeń z pierwszego kopca poprzedza na liście korzeń z drugiego kopca);

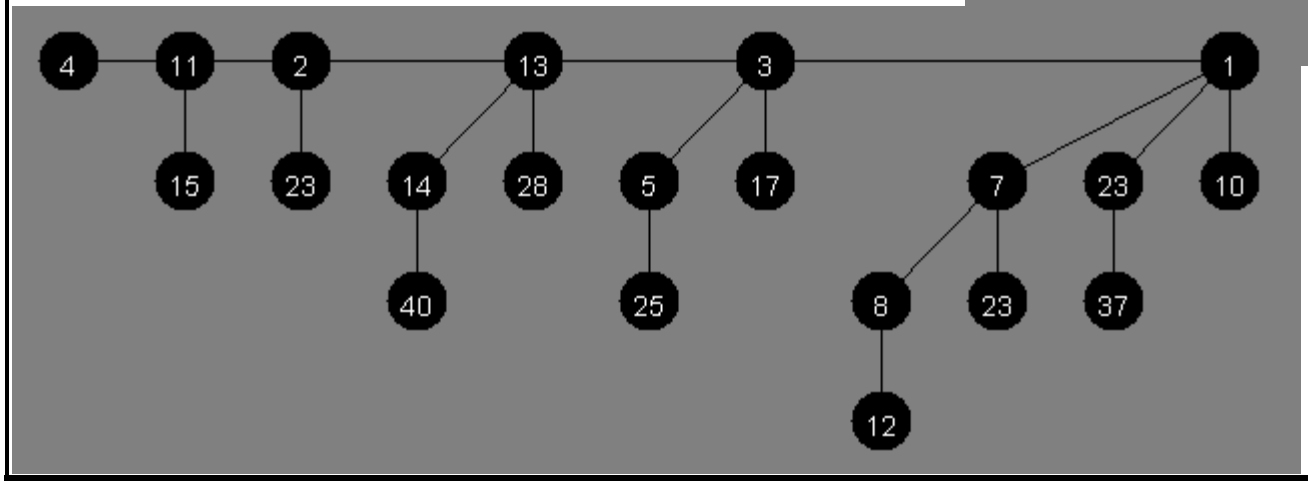
- rekurencyjnego łączenia drzew o tym samym stopniu korzenia, aż do przywrócenia podstawowej własności kopca - istnienia w kopcu co najwyżej jednego drzewa o korzeniu stopnia i-tego; wówczas ten korzeń, którego klucz ma wartość większą, staje się lewym skrajnym potomkiem drugiego korzenia; w ten sposób z dwóch drzew B_i powstaje jedno drzewo B_{i+1}.

Podczas operacji łączenia drzew może się zdarzyć, że trzy kolejne drzewa na liście mają korzenie tego samego stopnia. Wówczas łączy się te, których korzenie na liście położone są „dalej” od początku listy.

Łączenie dwóch kopców dwumianowych

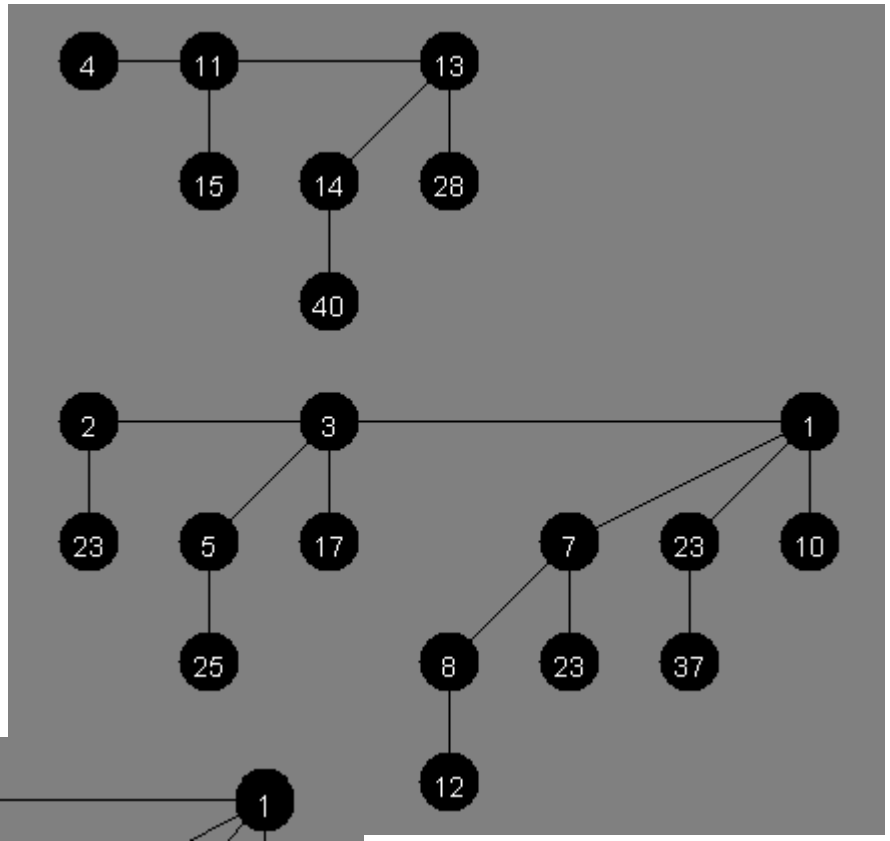
Przykład

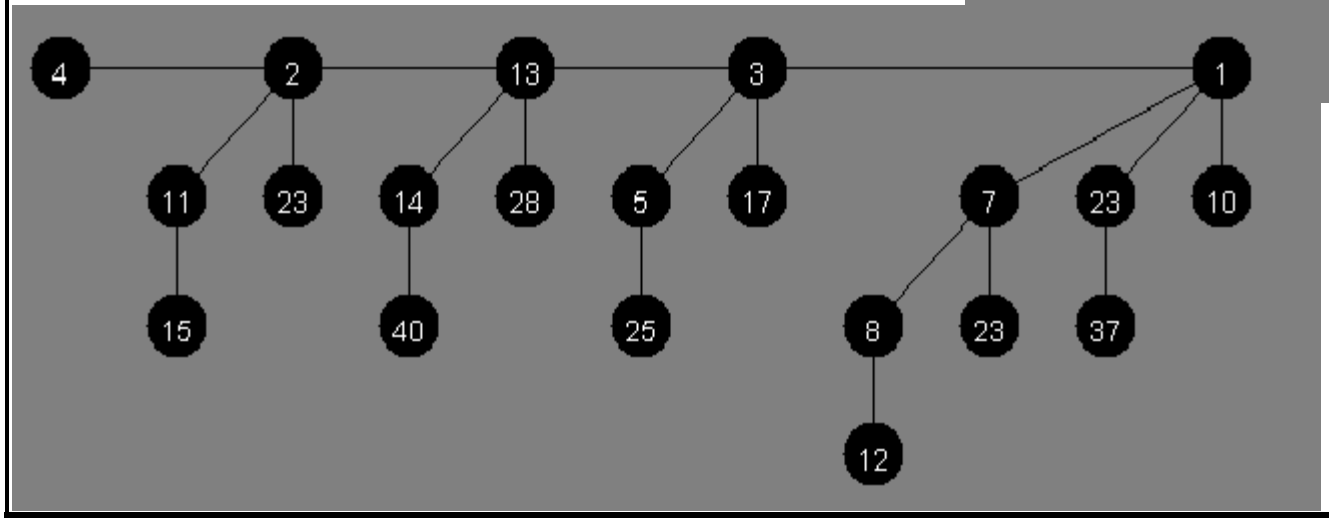




Łączenie dwóch kopców dwumianowych

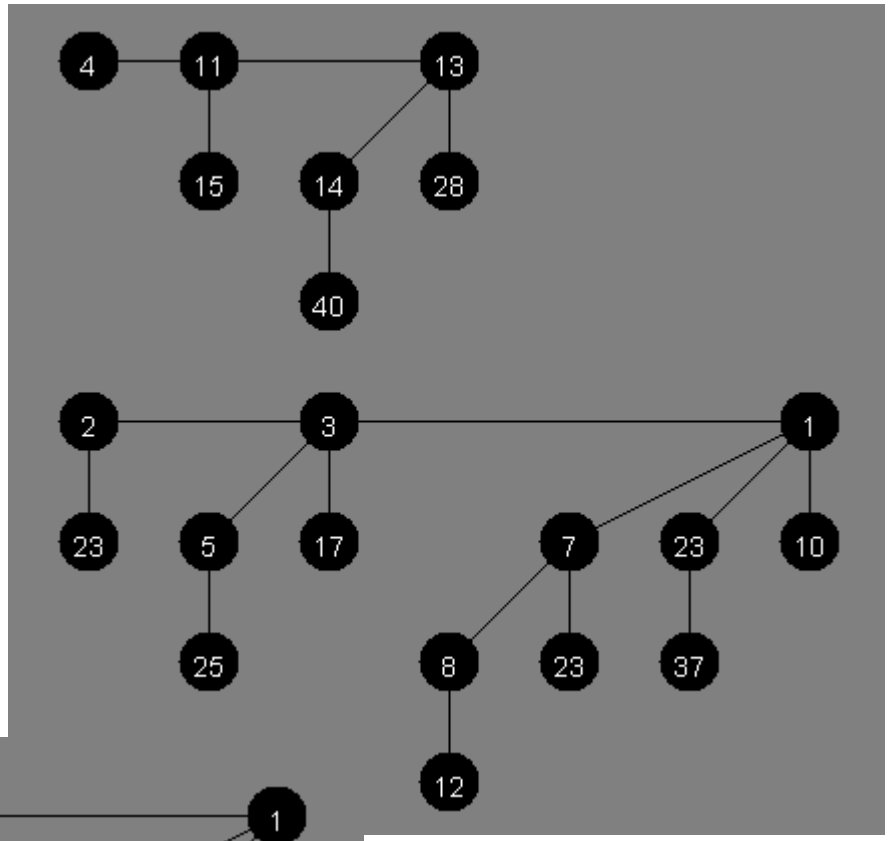
Przykład

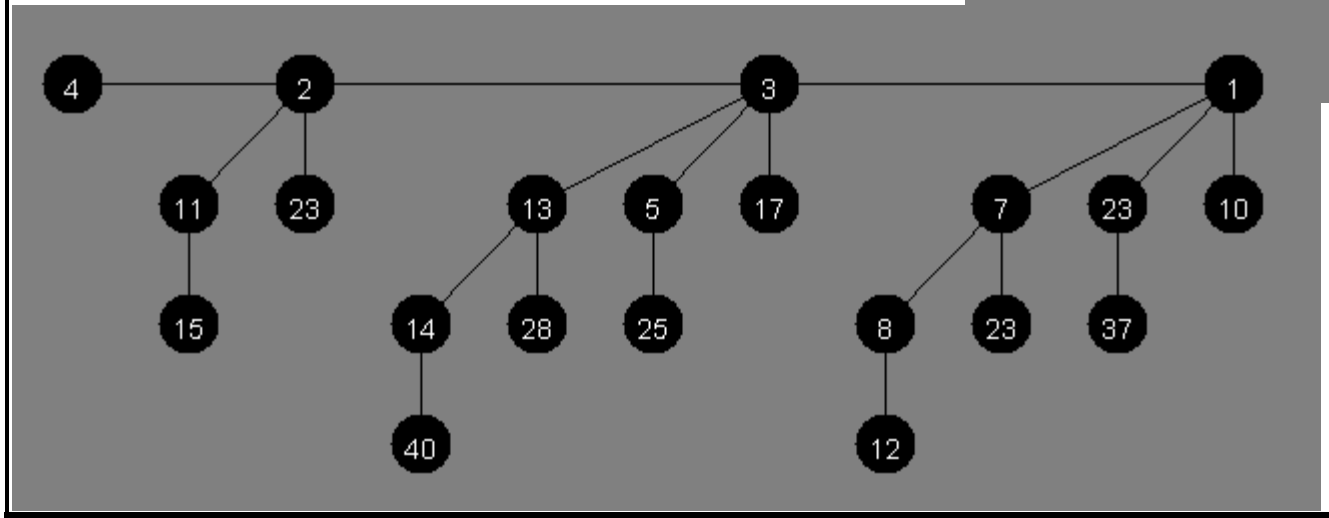




Łączenie dwóch kopców dwumianowych

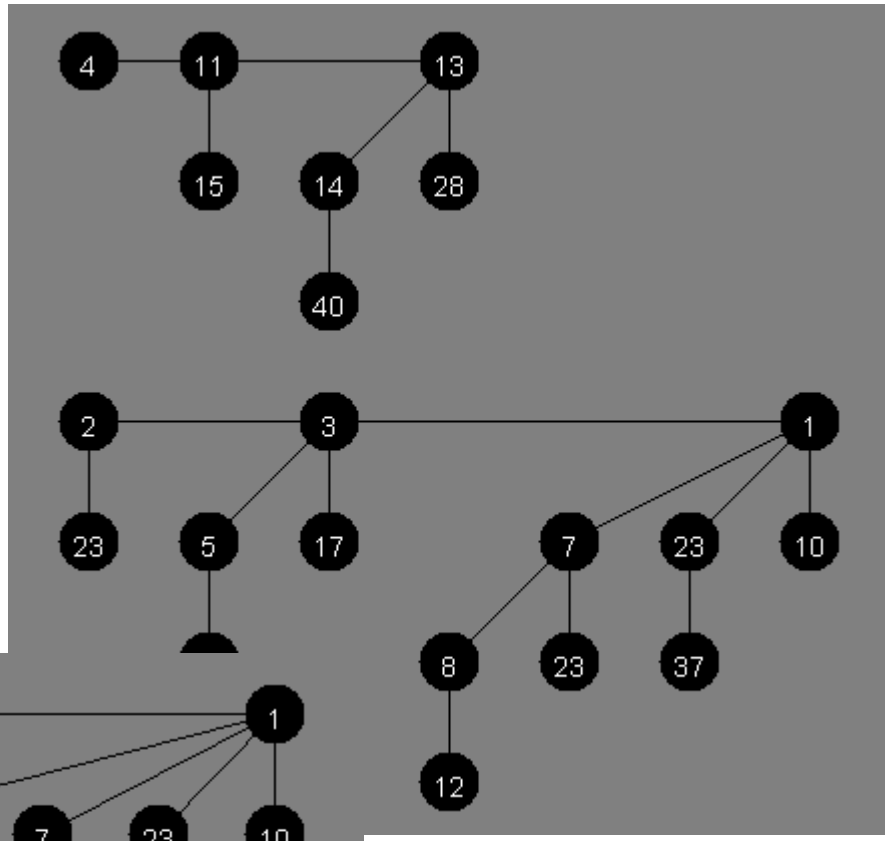
Przykład



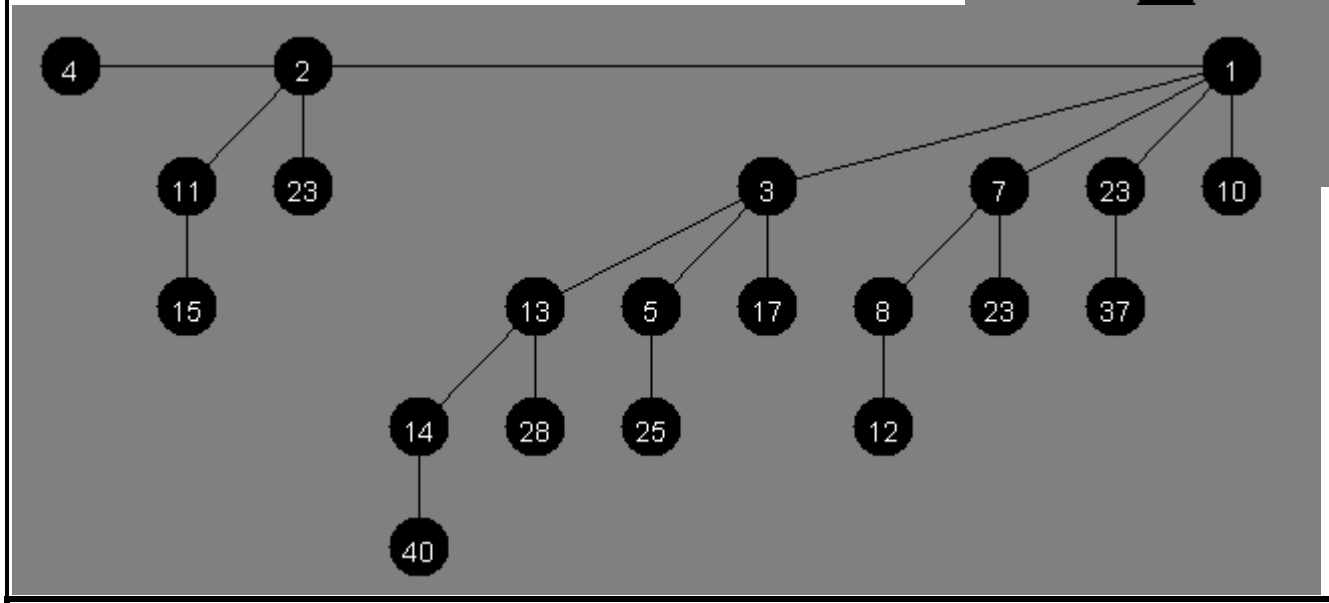


Łączenie dwóch kopców dwumianowych

Przykład



Złożoność $O(\lg N)$

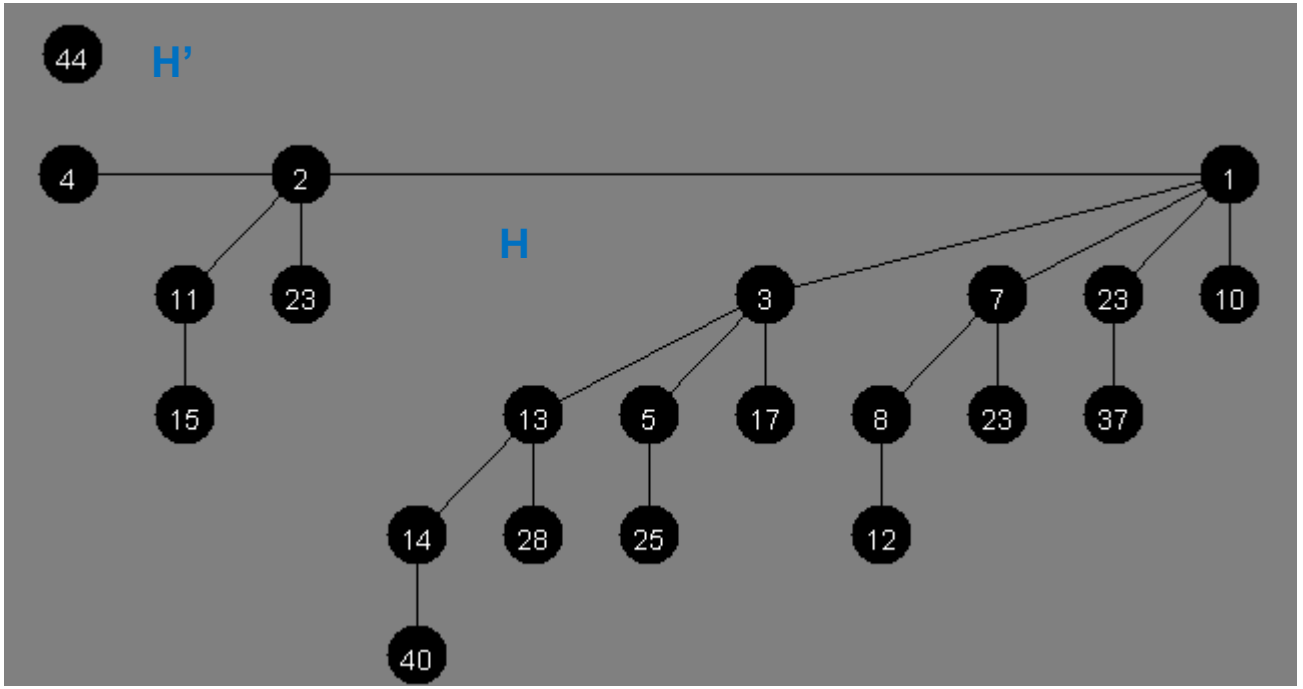


Wstawianie węzła

Operacja wstawiania nowego węzła *x* do kopca H składa się z dwóch faz:

- **utworzenia nowego kopca H', zawierającego jedynie wstawiany nowy węzeł *x* jako drzewo dwumianowe B_0;**
- **połączenia kopców H i H'.**

Przykład - wstawienie węzła o kluczu 44.



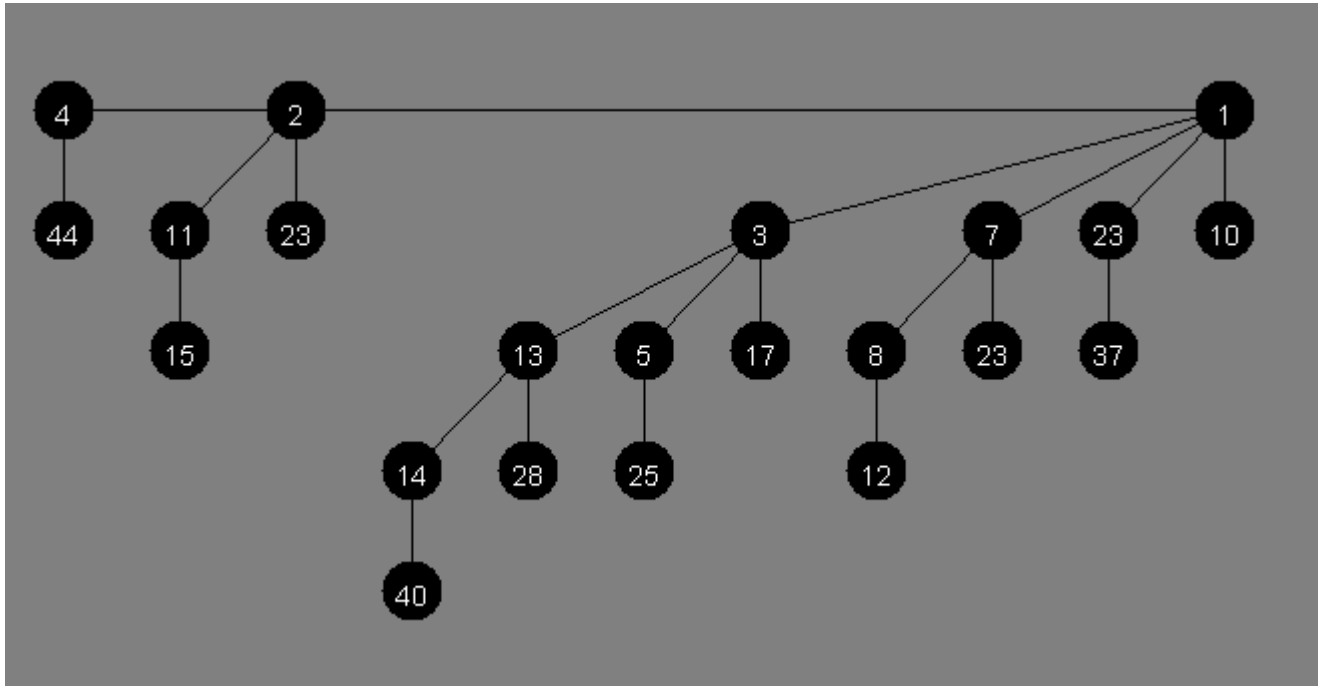
Wstawianie węzła

Operacja wstawiania nowego węzła *x* do kopca H składa się z dwóch faz:

- **utworzenia nowego kopca H', zawierającego jedynie wstawiany nowy węzeł *x* jako drzewo dwumianowe B_0;**
- **połączenia kopców H i H'.**

Złożoność $O(\lg N)$

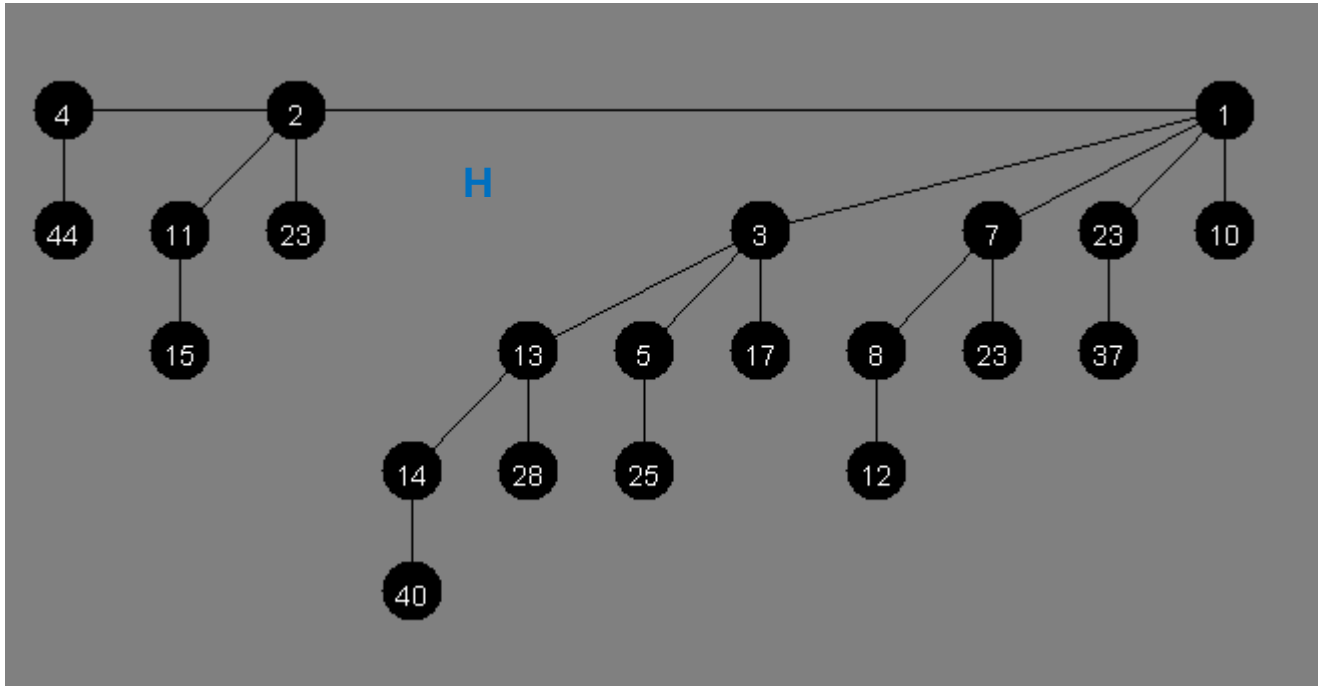
Przykład - wstawienie węzła o kluczu 44.



Usuwanie węzła o najmniejszej wartości klucza

Operacja składa się z trzech faz:

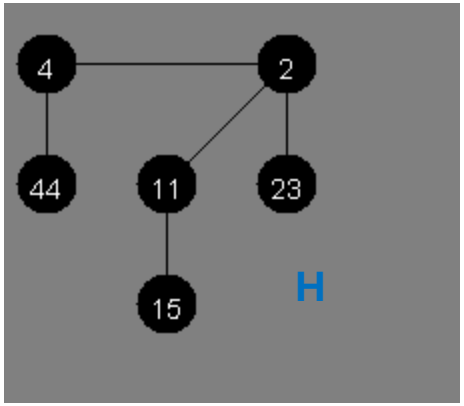
- **znalezienia na liście korzeni drzew składowych kopca H węzła o najmniejszej wartości klucza i usunięcia odpowiadającego mu drzewa z kopca H;**
- **utworzenia nowego kopca H', którego drzewami będą drzewa dwumianowe o korzeniach będących potomkami usuwanego węzła (kolejność potomków należy odwrócić);**
- **połączenia kopców H i H'.** **Przykład**

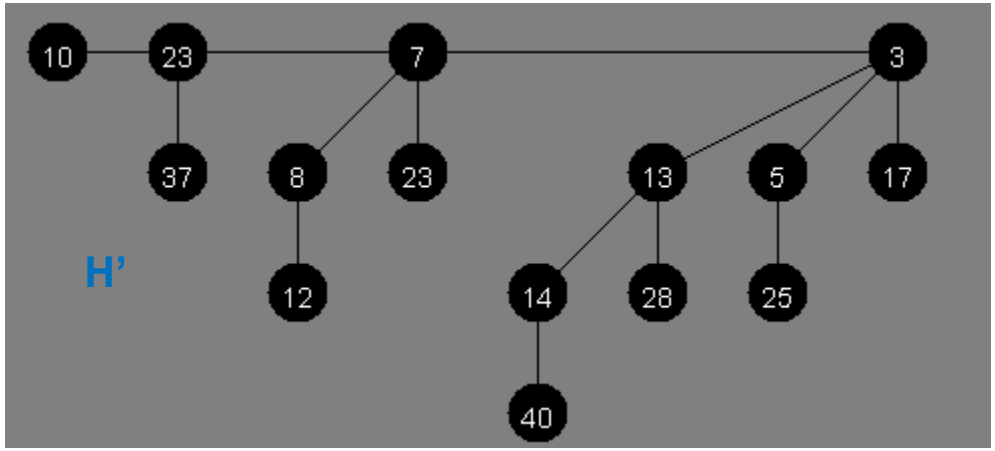


Usuwanie węzła o najmniejszej wartości klucza

Operacja składa się z trzech faz:

- **znalezienia na liście korzeni drzew składowych kopca H węzła o najmniejszej wartości klucza i usunięcia odpowiadającego mu drzewa z kopca H;**
- **utworzenia nowego kopca H', którego drzewami będą drzewa dwumianowe o korzeniach będących potomkami usuwanego węzła (kolejność potomków należy odwrócić);**
- **połączenia kopców H i H'.** **Przykład**





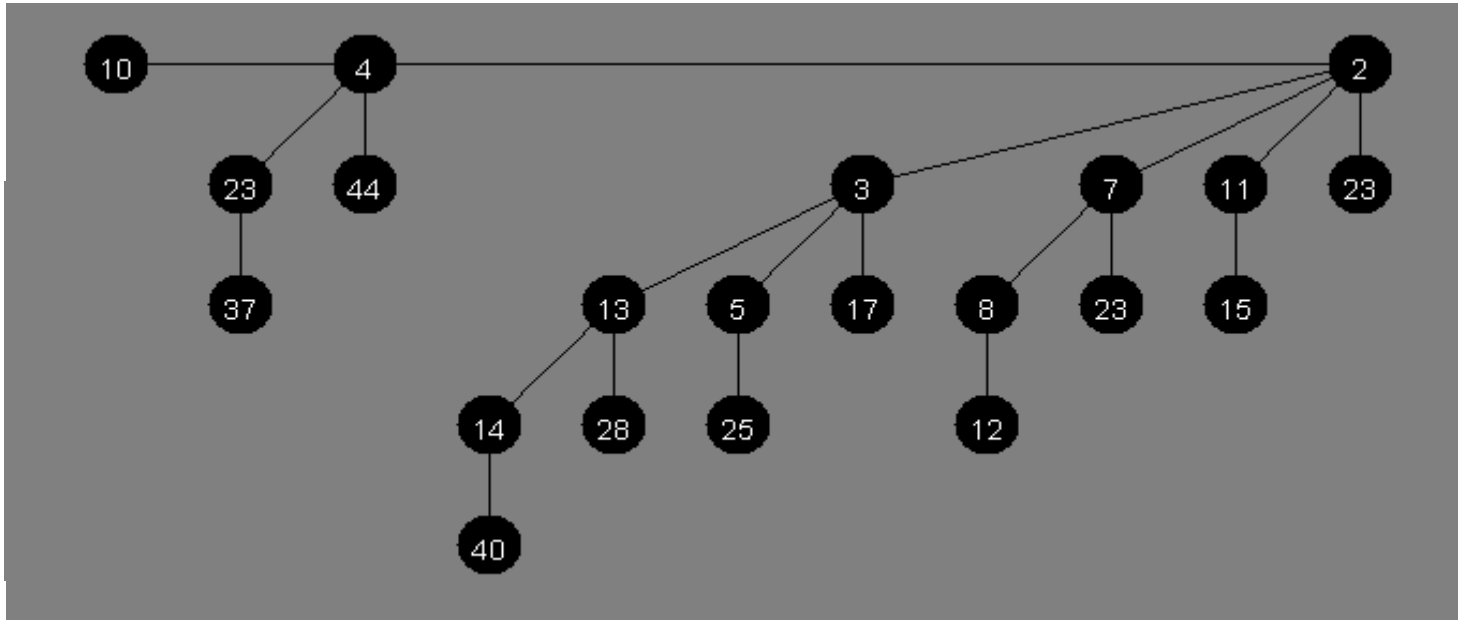
Usuwanie węzła o najmniejszej wartości klucza

Operacja składa się z trzech faz:

- **znalezienia na liście korzeni drzew składowych kopca H węzła o najmniejszej wartości klucza i usunięcia odpowiadającego mu drzewa z kopca H;**
- **utworzenia nowego kopca H', którego drzewami będą drzewa dwumianowe o korzeniach będących potomkami usuwanego węzła (kolejność potomków należy odwrócić);**

Złożoność $O(\lg N)$

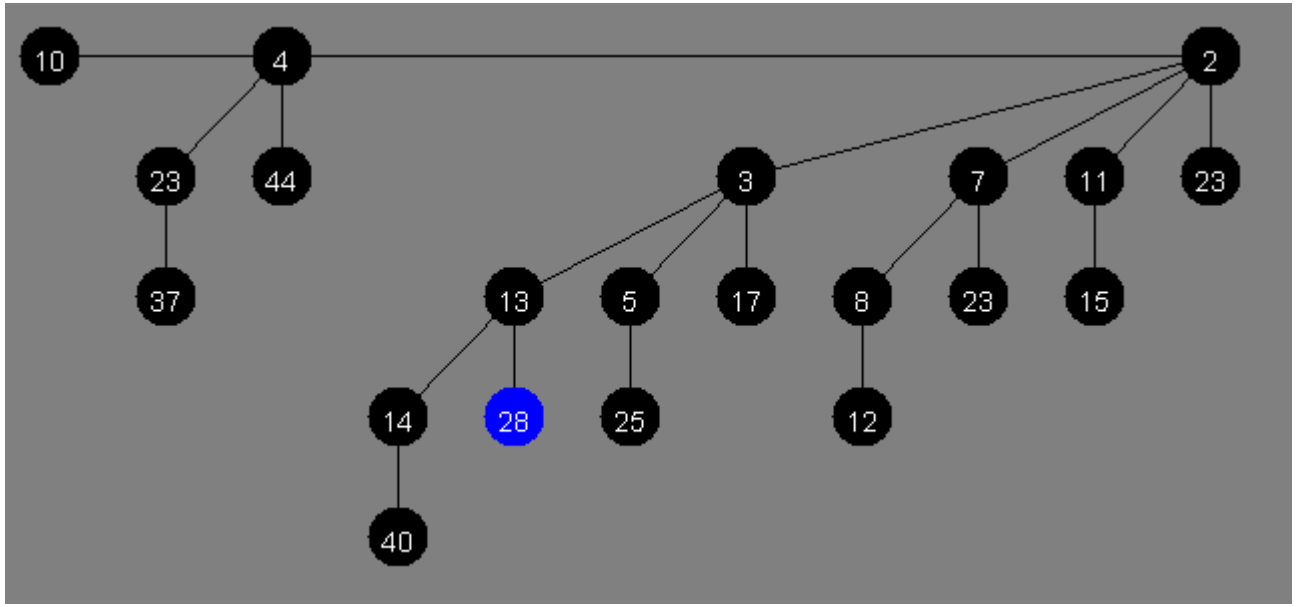
- **połączenia kopców H i H'.** **Przykład**



Zmniejszanie wartości klucza w dowolnym węźle

Operacja, podobnie jak w klasycznych kopcach binarnych, polega na „promowaniu” w kierunku korzenia (przez zamianę miejscami potomka i przodka) węzła naruszającego zasadę uporządkowania kopca.

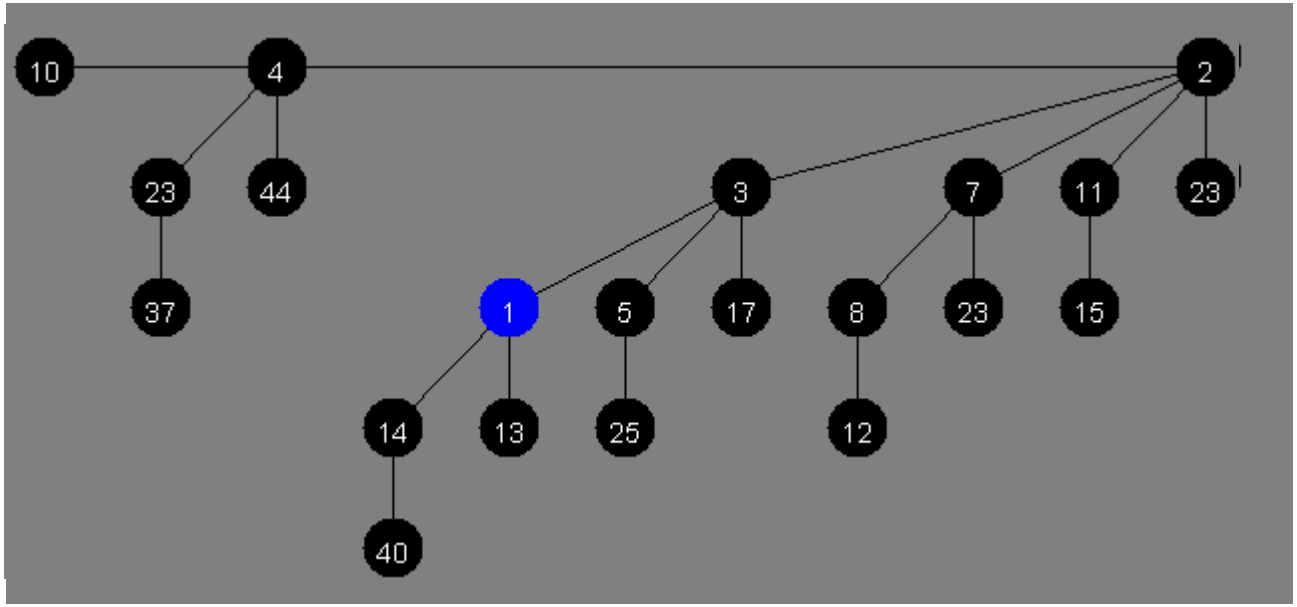
Przykład - zmniejszenie wartości klucza wskazanego węzła z 28 na 1.



Zmniejszanie wartości klucza w dowolnym węźle

Operacja, podobnie jak w klasycznych kopcach binarnych, polega na „promowaniu” w kierunku korzenia (przez zamianę miejscami potomka i przodka) węzła naruszającego zasadę uporządkowania kopca.

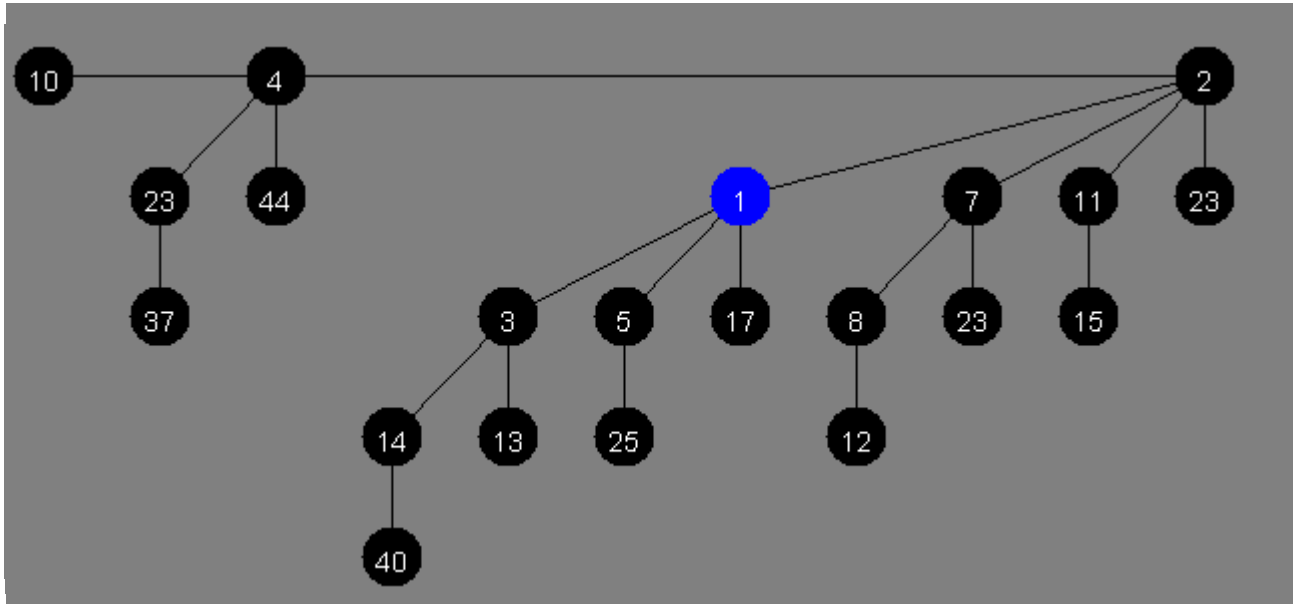
Przykład - zmniejszenie wartości klucza wskazanego węzła z 28 na 1.



Zmniejszanie wartości klucza w dowolnym węźle

Operacja, podobnie jak w klasycznych kopcach binarnych, polega na „promowaniu” w kierunku korzenia (przez zamianę miejscami potomka i przodka) węzła naruszającego zasadę uporządkowania kopca.

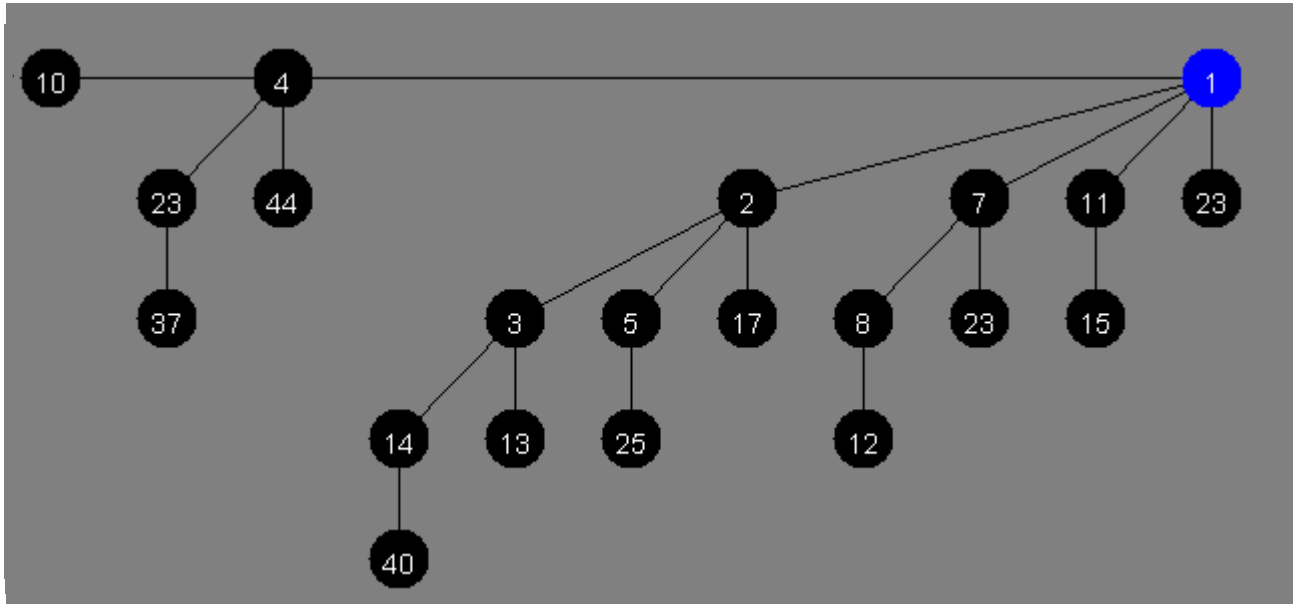
Przykład - zmniejszenie wartości klucza wskazanego węzła z 28 na 1.



Zmniejszanie wartości klucza w dowolnym węźle

Operacja, podobnie jak w klasycznych kopcach binarnych, polega na „promowaniu” w kierunku korzenia (przez zamianę miejscami potomka i przodka) węzła naruszającego zasadę uporządkowania kopca.

Przykład - zmniejszenie wartości klucza wskazanego węzła z 28 na 1.



Złożoność $O(\lg N)$

Koniec części 7