

## Algorytmy i struktury danych (Lista 8)

### Zadanie 1

1. Czym różni się haszowanie łańcuchowe od otwartego?  
Łańcuchowe: elementy o danym indeksie są przechowywane w liście pod danym indeksem w tablicy (czasem drzewie)  
Otwarte: elementy trafiają bezpośrednio do tablicy, ale pod obliczany indeks
2. Czym różnią się dwie wersje haszowania otwartego: haszowanie liniowe i haszowanie podwójne?  
Liniowe -> pierwsza wolna komórka  
Podwójne -> po wykryciu kolizji przy  $h()$  sprawdza  $h() + h2()$  [wielokrotności  $h2$  dodaje] mod  $n$
3. Dla tablicy z haszowaniem podwójnym o rozmiarze  $m = 11$  i funkcjach haszujących:  $h1(x) = x \bmod 11$  oraz  $h2(x) = x \bmod 10 + 1$  wyznacz ciąg kontrolny dla liczby 23. Jak wyglądałby ten ciąg w przypadku haszowania liniowego?  
Podwójne: 1, 5, 9, 2, 6, 10, 3, 7, 0, 4, 8  
Liniowe: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 0

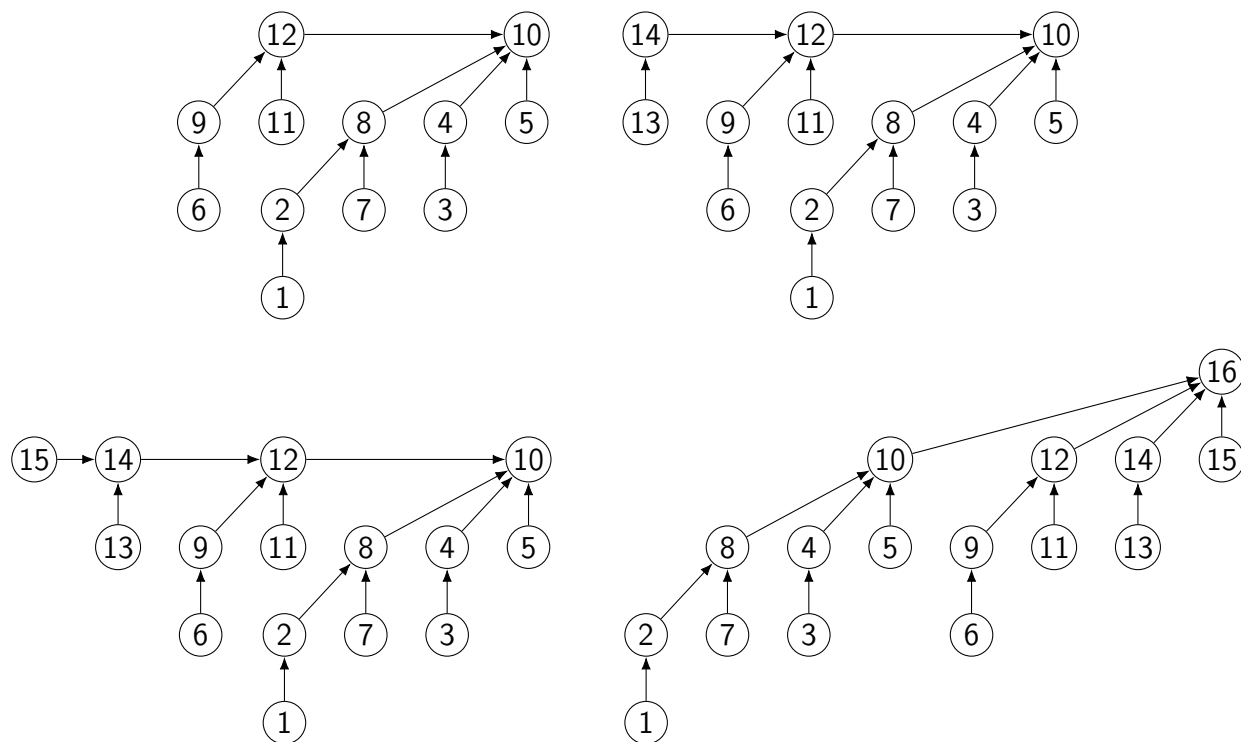
**Zadanie 2** (2pkt) Porównaj jaka będzie łączna liczba kolizji, gdy do tablicy z poprzedniego zadania wstawimy kolejno liczby: 22, 66, 44, 23, 35, używając:

- (a) haszowania liniowego: 7
- (b) haszowania dwukrotnego: 2
- (c) haszowania łańcuchowego: 2

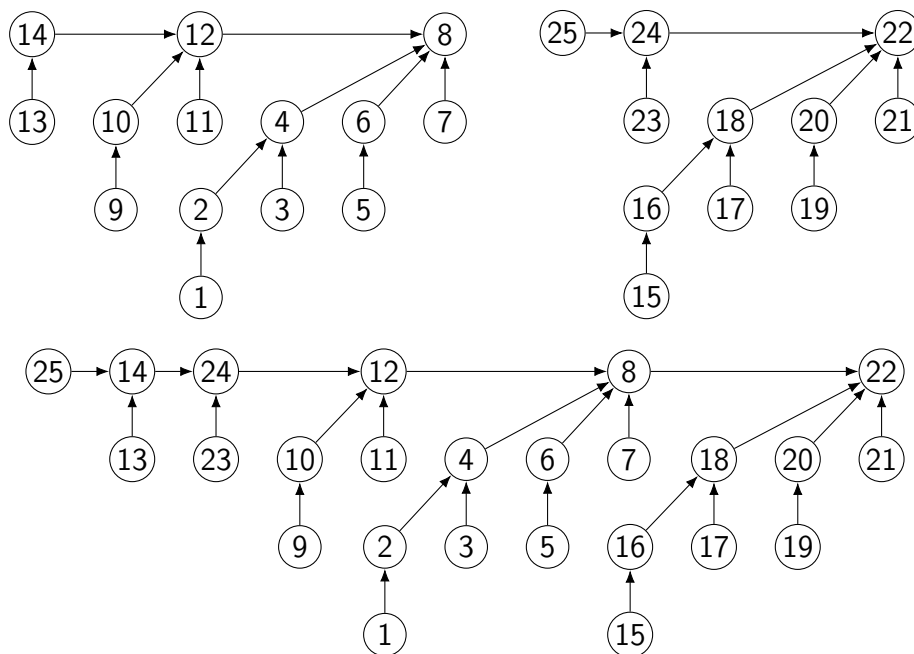
Następnie w każdym z wariantów sprawdź, jaka będzie łączna liczba porównań kluczy, gdy w gotowej tablicy wywołamy kolejno procedurę FIND (a) dla każdego elementu obecnego w tablicy, (b) dla elementów: 24 i 34, których nie ma w tablicy.

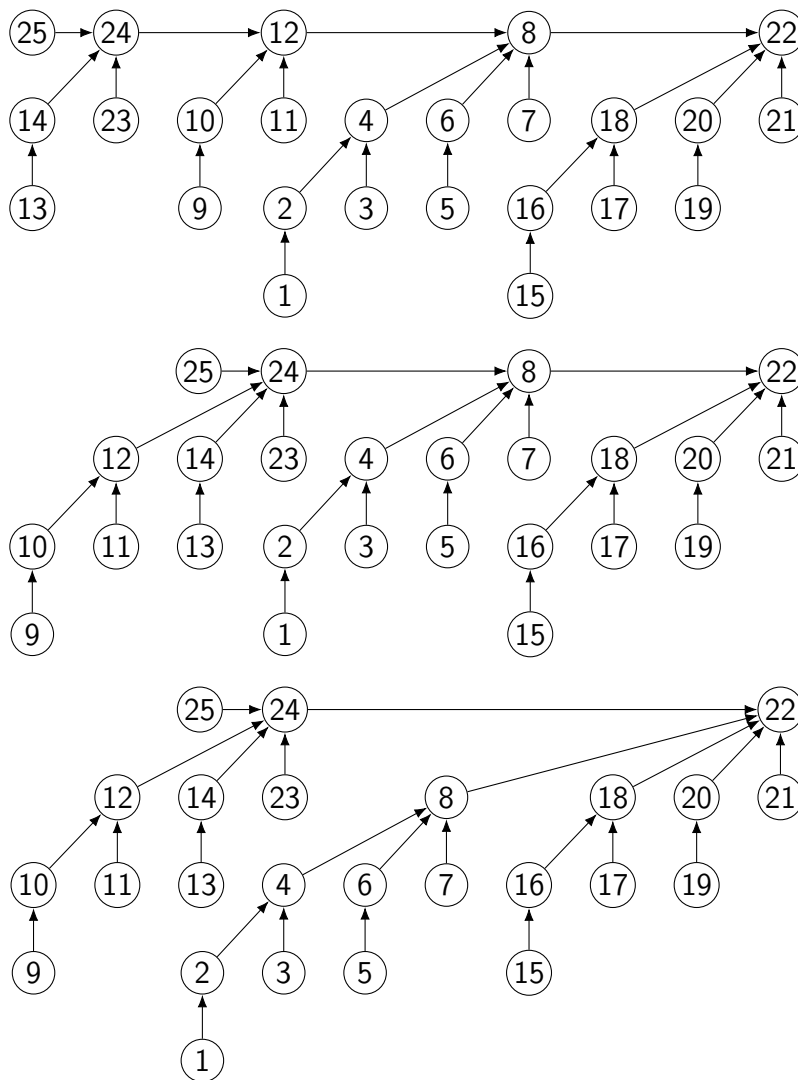
- (a) haszowania liniowego: a] 12 b] 7
- (b) haszowania dwukrotnego: a] 7 b] 4
- (c) haszowania łańcuchowego: a] 8 b] 2

**Zadanie 3** Narysuj przykładowe kopce dwumianowe o 12, 14, 15 i 16 węzłach. Na rysunku uwzględnij wartości kluczy oraz stopnie węzłów. W kopcu 12-elementowym zaznacz dodatkowo strzałki (najlepiej w różnych kolorach) przedstawiające wskaźniki na ojca, syna i brata.



**Zadanie 4** Zilustruj działanie operacji UNION łączącej kopiec dwumianowy o 14 węzłach z kopcem dwumianowym o 11 węzłach. Przyjmij dowolne wartości kluczy spełniające warunek kopca.



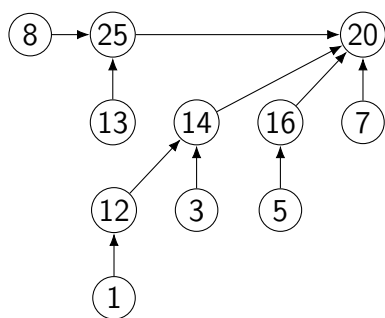


**Zadanie 6** Napisz funkcję `int ile_drzew_w_kopcu(int n)` wyliczającą ile jest drzew dwumianowych w kopcu dwumianowym zawierającym  $n$  kluczy.

```
int ile_drzew_w_kopcu(int n) {
    int i = 0; // zlicza 1 w bitowej reprezentacji n
    while (n > 0) {
        if (n % 2 == 1) ++i;
        n /= 2;
    }
    return i;
}
```

**Zadanie 7**

(a) Do pustego kopca dwumianowego wstaw (INSERT) kolejno: 1, 12, 3, 14, 5, 16, 7, 20, 25, 13, 8



(b) Dla otrzymanego kopca dwukrotnie wykonaj operację GETMAX (kradzież max).

