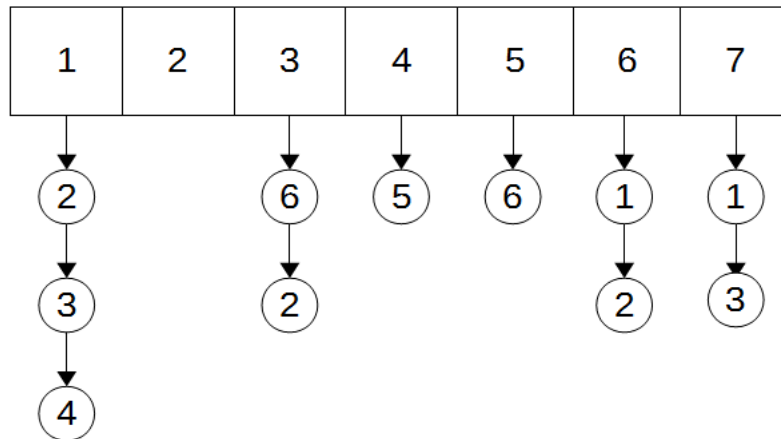


ASD, kolokwium A – 12 czerwca, 2017

- (15p) W strukturze słownikowej trzymamy elementy a, b, c, e, g, h. Wyszukiujemy następujące elementy z podanymi prawdopodobieństwami a(0,3), b(0,1), c(0,1), d(0,2), e(0,1), f(0,1), g(0), h(0,1). Stwórz optymalne dla wyszukiwań drzewo BST o korzeniu c. Oblicz średnią ilość odwiedzonych elementów podczas wyszukiwania dla optymalnej listy nieuporządkowanej (5p) i dla optymalnego drzewa BST o wartości korzenia c (5p).
- Narysuj graf skierowany reprezentowany przez poniższą strukturę (5p). Narysuj drzewo wywołań rekurencyjnych dla funkcji funkcji visit_dfs(1) (10p).



```
visit_dfs(x){
    if(mark[x]==visited) return;
    mark[x]=visited;
    Edge *ptr=edges[x];
    while(ptr!=NULL){
        if(mark[ptr->end]!=visited) visit_dfs(ptr->end);
        ptr=ptr->next;
    }
}
```

- (15p) Napisz funkcję `int Roznica(Node *ptr, int x)`, która zwróci różnicę między ilością elementów o parzystych kluczach i ilością elementów o nieparzystych kluczach. Przyjmij, że `Node` to struktura postaci `struct Node{int key; Node *left; Node *right; }`.