Stromy

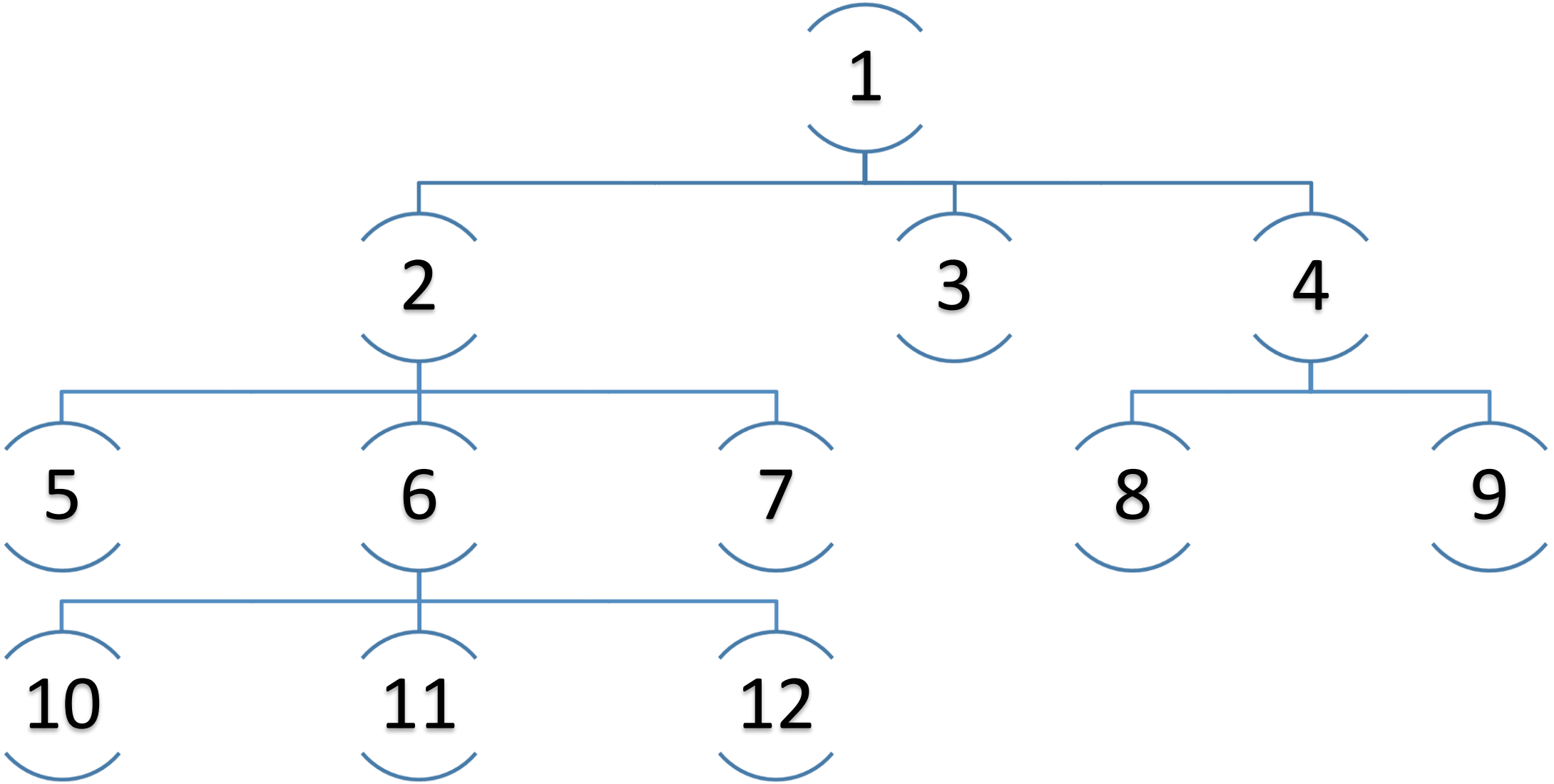
Jiří Zacpal

KATEDRA INFORMATIKY
UNIVERZITA PALACKÉHO V OLOMOUCI

KMI/ZADS - Základní algoritmy a datové struktury

Strom

Příklad



Realizace stromu v Pythonu

```
root={"id":1,"children":[
      {"id":2,"children":[
            {"id":5,"children":[],"n":0},
            {"id":6,"children":[
                  {"id":10,"children":[],"n":0},
                  {"id":11,"children":[],"n":0},
                  {"id":12,"children":[],"n":0}],"n":3},
            {"id":7,"children":[],"n":0}],"n":3},
      {"id":3,"children":[],"n":0},
      {"id":4,"children":[
            {"id":8,"children":[],"n":0},
            {"id":9,"children":[],"n":0}],"n":2}],"n":3}
```

Průchod do hloubky

Průchod do hloubky - rekurzivně

```
def depth_first_search(node):
    x=node
    print(x["id"], end=" ")
    for i in range(x["n"]):
        depth_first_search(x["children"][i])
```

Průchod do hloubky - iteračně

```
def depth_first_search_iter(node):
    x=node
    S=lds.init_stack(20)
    lds.push(S,x)
    while lds.empty_s(S)!=True:
        y=lds.pop(S)
        print(y["id"],end=" ")
        for i in range(y["n"]):
            lds.push(S,y["children"][i])
```

Průchod do šířky

Průchod do šířky

```
def breadth_first_search_iter(node):
    x=node
    Q=lds.init_queue(20)
    lds.enqueue(Q,x)
    while lds.empty_q(Q)!=True:
        y=lds.dequeue(Q)
        print(y["id"],end=" ")
        for i in range(y["n"]):
            lds.enqueue(Q,y["children"][i])
```

Úkol

1. Implementujte strom pomocí struktury, kdy jsou potomci v seznamu:

   struct node

   id : key

   child : node

   sibling : node.

2. Napište funkce pro průchod do hloubky tímto stromem.