MODUL PRAKTIKUM

ALGORITMA DAN STRUKTUR DATA

INF1008

Penyusun :

Naufal Azmi Verdikha, M.Eng.

Teknik Informatika
Fakultas Sains & Teknologi
Universitas Muhammadiyah Kalimantan Timur

Samarinda, 2018

Praktikum 11:
Algoritma Pohon

Pokok Bahasan:

- ❖ Algoritma Pohon
- ❖ Binary Heap

Tujuan Pembelajaran:

- ✓ Memahami implementasi algoritma pohon menggunakan struktur list, kelas dan referensi
- ✓ Memahami implementasi operasi binary heap

Representasi List:

Percobaan & Latihan 11.1

Jalankan kedua program berikut!

```
myTree = ['a', ['b', ['d',[],[]], ['e',[],[]] ], ['c', ['f',[],[]], []] ]
print(myTree)
print('left subtree = ', myTree[1])
print('root = ', myTree[0])
print('right subtree = ', myTree[2])

```

```
def BinaryTree(r):
    return [r, [], []]

def insertLeft(root,newBranch):
    t = root.pop(1)
    if len(t) > 1:
        root.insert(1,[newBranch,t,[]])
    else:
        root.insert(1,[newBranch, [], []])
    return root

def insertRight(root,newBranch):
    t = root.pop(2)
    if len(t) > 1:
        root.insert(2,[newBranch,[],t])
    else:
        root.insert(2,[newBranch,[],[]])
    return root

def getRootVal(root):
    return root[0]

def setRootVal(root,newVal):
    root[0] = newVal
```

```
25
26  def getLeftChild(root):
27      return root[1]
28
29  def getRightChild(root):
30      return root[2]
31
32  r = BinaryTree(3)
33  insertLeft(r,4)
34  insertLeft(r,5)
35  insertRight(r,6)
36  insertRight(r,7)
37  l = getLeftChild(r)
38  print(l)
39
40  setRootVal(l,9)
41  print(r)
42  insertLeft(l,11)
43  print(r)
44  print(getRightChild(getRightChild(r)))
45
```

Soal:

a) Berikan tampilan ouput dari kedua program diatas!
b) Jelaskan kegunaan fungsi-fungsi yang terdapat di program kedua!

Node dan Referensi:

Percobaan & Latihan 11.2

Jalan program berikut!

```
1  class BinaryTree:
2      def __init__(self,rootObj):
3          self.key = rootObj
4          self.leftChild = None
5          self.rightChild = None
6
7      def insertLeft(self,newNode):
8          if self.leftChild == None:
9              self.leftChild = BinaryTree(newNode)
10         else:
11             t = BinaryTree(newNode)
12             t.leftChild = self.leftChild
13             self.leftChild = t
14
15     def insertRight(self,newNode):
16         if self.rightChild == None:
17             self.rightChild = BinaryTree(newNode)
18         else:
19             t = BinaryTree(newNode)
20             t.rightChild = self.rightChild
21             self.rightChild = t
22
23
24     def getRightChild(self):
25         return self.rightChild
26
27     def getLeftChild(self):
28         return self.leftChild
29
30     def setRootVal(self,obj):
31         self.key = obj
```

```
32
33      def getRootVal(self):
34          return self.key
35
36
37  r = BinaryTree('a')
38  print(r.getRootVal())
39  print(r.getLeftChild())
40  r.insertLeft('b')
41  print(r.getLeftChild())
42  print(r.getLeftChild().getRootVal())
43  r.insertRight('c')
44  print(r.getRightChild())
45  print(r.getRightChild().getRootVal())
46  r.getRightChild().setRootVal('hello')
47  print(r.getRightChild().getRootVal())
48
```

Soal :

a) Berikan tampilan output dari program diatas!
b) Jelaskan kegunaan kelas BinaryTree pada program diatas!

Pohon Parse:

Percobaan & Latihan 11.3

Jalankan program berikut!

```
1  from pythonds.basic.stack import Stack
2  from pythonds.trees.binaryTree import BinaryTree
3
4  def buildParseTree(fpexp):
5      fplist = fpexp.split()
6      pStack = Stack()
7      eTree = BinaryTree('')
8      pStack.push(eTree)
9      currentTree = eTree
10     for i in fplist:
11         if i == '(':
12             currentTree.insertLeft('')
13             pStack.push(currentTree)
14             currentTree = currentTree.getLeftChild()
15         elif i not in ['+', '-', '*', '/', ')']:
16             currentTree.setRootVal(int(i))
17             parent = pStack.pop()
18             currentTree = parent
19         elif i in ['+', '-', '*', '/']:
20             currentTree.setRootVal(i)
21             currentTree.insertRight('')
22             pStack.push(currentTree)
23             currentTree = currentTree.getRightChild()
24         elif i == ')':
25             currentTree = pStack.pop()
26         else:
27             raise ValueError
28     return eTree
29
30 pt = buildParseTree("( ( 10 + 5 ) * 3 )")
31 pt.postorder()  #defined and explained in the next section
32
```

Soal:

a) Berikan tampilan output pada program diatas!
b) Pada baris keberapa penggunaan algoritma stack pada program diatas?
c) Jelaskan kegunaan fungsi buildParseTree pada program diatas!

Implementasi Operasi Binary Heap:

Percobaan & Latihan 11.4

Jalankan program berikut!

```
class BinHeap:
    def __init__(self):
        self.heapList = [0]
        self.currentSize = 0


    def percUp(self,i):
        while i // 2 > 0:
          if self.heapList[i] < self.heapList[i // 2]:
             tmp = self.heapList[i // 2]
             self.heapList[i // 2] = self.heapList[i]
             self.heapList[i] = tmp
          i = i // 2

    def insert(self,k):
      self.heapList.append(k)
      self.currentSize = self.currentSize + 1
      self.percUp(self.currentSize)

    def percDown(self,i):
      while (i * 2) <= self.currentSize:
          mc = self.minChild(i)
          if self.heapList[i] > self.heapList[mc]:
              tmp = self.heapList[i]
              self.heapList[i] = self.heapList[mc]
              self.heapList[mc] = tmp
          i = mc

```

```
28
29      def minChild(self,i):
30        if i * 2 + 1 > self.currentSize:
31            return i * 2
32        else:
33            if self.heapList[i*2] < self.heapList[i*2+1]:
34                return i * 2
35            else:
36                return i * 2 + 1
37
38      def delMin(self):
39        retval = self.heapList[1]
40        self.heapList[1] = self.heapList[self.currentSize]
41        self.currentSize = self.currentSize - 1
42        self.heapList.pop()
43        self.percDown(1)
44        return retval
45
46      def buildHeap(self,alist):
47        i = len(alist) // 2
48        self.currentSize = len(alist)
49        self.heapList = [0] + alist[:]
50        while (i > 0):
51            self.percDown(i)
52            i = i - 1
53
54  bh = BinHeap()
55  bh.buildHeap([9,5,6,2,3])
56
57  print(bh.delMin())
58  print(bh.delMin())
59  print(bh.delMin())
60  print(bh.delMin())
61  print(bh.delMin())
62
```

Soal:

a) Berikan tampilan output pada program diatas!
b) Berikan keterangan penjelasan pada baris ke 46 hingga 52 pada program diatas!
c) Lakukanlah uji coba dengan item yang berbeda pada baris ke-55 pada program diatas dan berikan hasil output dan analisa dari uji coba!

Laporan Resmi:

1. Buatlah summary dan analisa dari **Percobaan & Latihan** pada pratikum ini.
2. Berikan kesimpulan dari praktikum ini.