



ALGORITMA DAN STRUKTUR DATA

INF1083

LAPORAN PRAKTIKUM 9: Searching

Oleh :

Nama: Fajar Fauzi

Nim: 2211102441088

Teknik Informatika
Fakultas Sains & Teknologi
Universitas Muhammadiyah Kalimantan Timur

Samarinda, 2023

Laporan Praktikum 9:

Searching

Pokok Bahasan:

- ❖ Searching
- ❖ Hashing

Tujuan Pembelajaran:

- ✓ Memahami implementasi searching menggunakan Python
- ✓ Memahami implementasi dari metode-metode hashing

Searching

Percobaan & Latihan: 9.1

Jalankan perintah berikut!

```
1 print(15 in [3,5,2,4,1])
2 print(5 in [3,5,2,4,1])
3 print(10 in [3,5,2,4,1])
```

Soal :

- a) Berikan tampilan output dari perintah diatas!

Output:

```
False
True
False
```

- b) Jelaskan operator in pada perintah diatas!

Penjelasan:

Oprator in berfungsi untuk mengecek apakah suatu nilai termasuk dalam anggota list.

Percobaan & Latihan: 9.2

Jalankan fungsi dan perintah berikut!

```

1 def sequentialSearch(alist, item):
2     pos = 0
3     found = False
4
5     while pos < len(alist) and not found:
6         if alist[pos] == item:
7             found = True
8         else:
9             pos = pos+1
10    return found
11
12 testlist = [1, 2, 32, 8, 17, 19, 42, 13, 0]
13 print(sequentialSearch(testlist, 3))
14 print(sequentialSearch(testlist, 13))

```

```

1 def orderedSequentialSearch(alist, item):
2     pos = 0
3     found = False
4     stop = False
5     while pos < len(alist) and not found and not stop:
6         if alist[pos] == item:
7             found = True
8         else:
9             if alist[pos] > item:
10                stop = True
11            else:
12                pos = pos+1
13
14    return found
15
16 testlist = [0, 1, 2, 8, 13, 17, 19, 32, 42,]
17 print(orderedSequentialSearch(testlist, 3))
18 print(orderedSequentialSearch(testlist, 13))

```

Soal :

- a) Berikan tampilan output dari kedua fungsi tersebut!

Output:

Sequential Search

```
False
True
```

Ordered Sequential Search

```
False
True
```

- b) Analisa perbedaan kedua fungsi tersebut!

Analisa:

Dapat bahwa pada program kedua memiliki variabel stop yang berfungsi untuk mengakhiri pencarian jika nilai sudah ditemukan atau melebihi nilai yang sedang dicari program akan langsung menampilkan output. Berbeda dengan program satu yang tidak memiliki variabel stop sehingga program akan menyelesaikan semua pencarian sampai nilai akhir.

Percobaan & Latihan: 9.3

Jalankan fungsi dan perintah berikut!

```

1 def binarySearch(alist, item):
2     first = 0
3     last = len(alist)-1
4     found = False
5
6     while first<=last and not found:
7         midpoint = (first + last)//2
8         if alist[midpoint] == item:
9             found = True
10        else:
11            if item < alist[midpoint]:
12                last = midpoint-1
13            else:
14                first = midpoint+1
15
16        return found
17
18 testlist = [0, 1, 2, 8, 13, 17, 19, 32, 42,]
19 print(binarySearch(testlist, 3))
20 print(binarySearch(testlist, 13))

```

```

1 def binarySearch(alist, item):
2     if len(alist) == 0:
3         return False
4     else:
5         midpoint = len(alist)//2
6         if alist[midpoint]==item:
7             return True
8         else:
9             if item<alist[midpoint]:
10                return binarySearch(alist[:midpoint],item)
11            else:
12                return binarySearch(alist[midpoint+1:],item)
13
14 testlist = [0, 1, 2, 8, 13, 17, 19, 32, 42,]
15 print(binarySearch(testlist, 3))
16 print(binarySearch(testlist, 13))

```

Soal :

- a) Berikan tampilan output dari kedua fungsi tersebut!

Output:

Program 1

False
True

Program 2

False
True

- b) Jelaskan fungsi variabel midpoint dari kedua fungsi diatas!

Penjelasan:

Pada kedua program variabel midpoint memiliki fungsi sebagai index yang digunakan pada parameter alist yang dimana nilai index didapatkan dengan menggunakan oprator aritmatik yang ada dalam variabel midpoint.

Percobaan & Latihan: 9.4

Jalankan perintah berikut!

```
1 print("Nilai ordinal dari karakter c adalah", ord('c'))
2 print("Nilai ordinal dari karakter a adalah", ord('a'))
3 print("Nilai ordinal dari karakter r adalah", ord('r'))
4 print("Nilai ordinal dari karakter i adalah", ord('i'))
```

Soal :

- a) Berikan tampilan output dari perintah tersebut!

Output:

```
Nilai ordinal dari karakter c adalah 99
Nilai ordinal dari karakter a adalah 97
Nilai ordinal dari karakter r adalah 114
Nilai ordinal dari karakter i adalah 105
```

- b) Jelaskan fungsi operator ord() diatas!

Penjelasan:

Fungsi dari ord() akan mengembalikan unicode dari setiap karakter yang dimasukan.

- c) Uji coba dengan menggunakan tanda baca, apakah tanda baca memiliki nilai ordinal?

Berikan tampilan output!

Output:

```
Nilai ordinal dari karakter . adalah 33
Nilai ordinal dari karakter ? adalah 63
```

Hashing

Percobaan & Latihan: 9.5

Jalankan fungsi dan perintah berikut!

```
1 def hash(astring, tablesize):
2     sum = 0
3     for pos in range(len(astring)):
4         sum = sum + ord(astring[pos])
5
6     return sum%tablesize
7
8 MyString = "cari"
9 print(hash(MyString,11))
```

Soal :

- a) Berikan tampilan output dari perintah tersebut!

Output:

```
8
```

- b) Jelaskan fungsi hash() diatas!

Penjelasan:

Fungsi program hash diatas akan menjumlahkan setiap unicode dari kata 'cari' kemudian setelah mendapatkan nilai dari setiap huruf akan dibagi dengan nilai

tablesize yaitu 11. Karena menggunakan pembagian modulo maka hasil akan dibulatkan ke angka yang paling mendekati yaitu 8.

- c) Uji coba dengan mengubah nilai variable MyString dengan tanda baca, apakah tanda baca memiliki slot ditabel hash? Berikan tampilan output!

Jawaban:

2 adalah output yang diperoleh dengan variabel mystring adalah tanda baca ‘.’.

Percobaan & Latihan: 9.6

Jalankan program berikut!

```

1 class HashTable:
2     def __init__(self):
3         self.size = 11
4         self.slots = [None] * self.size
5         self.data = [None] * self.size
6
7     def put(self, key, data):
8         hashvalue = self.hashfunction(key, len(self.slots))
9
10        if self.slots[hashvalue] == None:
11            self.slots[hashvalue] = key
12            self.data[hashvalue] = data
13        else:
14            if self.slots[hashvalue] == key:
15                self.data[hashvalue] = data #replace
16            else:
17                nextslot = self.rehash(hashvalue, len(self.slots))
18                while self.slots[nextslot] != None and \
19                    self.slots[nextslot] != key:
20                    nextslot = self.rehash(nextslot, len(self.slots))
21
22                if self.slots[nextslot] == None:
23                    self.slots[nextslot] = key
24                    self.data[nextslot] = data
25                else:
26                    self.data[nextslot] = data #replace
27
28    def hashfunction(self, key, size):
29        return key % size
30
31    def rehash(self, oldhash, size):
32        return (oldhash + 1) % size
33
34    def get(self, key):
35        startslot = self.hashfunction(key, len(self.slots))
36
37        data = None
38        stop = False
39        found = False
40        position = startslot
41        while self.slots[position] != None and \
42            not found and not stop:
43            if self.slots[position] == key:
44                found = True
45                data = self.data[position]

```

```

46         else:
47             position=self.rehash(position,len(self.slots))
48             if position == startslot:
49                 stop = True
50             return data
51
52     def __getitem__(self,key):
53         return self.get(key)
54
55     def __setitem__(self,key,data):
56         self.put(key,data)
57
58 H=HashTable()
59 H[54]="cat"
60 H[26]="dog"
61 H[93]="lion"
62 H[17]="tiger"
63 H[77]="bird"
64 H[31]="cow"
65 H[44]="goat"
66 H[55]="pig"
67 H[20]="chicken"
68 print(H.slots)
69 print(H.data)
70
71 print(H[20])
72
73 print(H[17])
74 H[20]='duck'
75 print(H[20])
76 print(H[99])

```

Soal :

- a) Berikan tampilan output dari perintah tersebut!

Output:

```

[77, 44, 55, 20, 26, 93, 17, None, None, 31, 54]
['bird', 'goat', 'pig', 'chicken', 'dog', 'lion', 'tiger', None, None, 'cow', 'cat']
tiger
duck
None

```

- b) Jelaskan fungsi kelas HashTable diatas!
Membuat sebuah penyimpanan data yang memiliki kunci dan nilai agar dapat melakukan pencarian dengan lebih cepat dan efisien.
- c) Berikah penjelasan pada baris 67, 68 dan 69 pada program diatas!

Penjelasan:

NO	Code	Penjelasan
67	h[20]='chiken'	Memasukan string data “chiken” kedalam slot 20
68	print(h.slots)	Melihat isi slot
69	print(h.data)	Menampilkan data pada table hash

Kesimpulan:

Setelah melakukan praktikum pada materi Searching ada beberapa point yang dapat saya simpulkan.

Dapat disimpulkan bahwa searching adalah proses mencari atau menemukan data atau informasi yang dibutuhkan dari dari sebuah kumpulan data yang disimpan. Dengan cara mengecek semua data dan mencocoknya dengan data yang sedang dicari atau melihat urutan nomor kunci yang menyimpan data atau informasi tersebut. Untuk mempermudah pencarian data dalam skala kecil atau besar.

Hashing merupakan proses mengubah data atau pesan menjadi nilai hash atau nilai acak tetap yang unik yang dimana nilai atau pesan yang akan diubah kedalam bentuk angka atau karakter simbol-simbol tertentu. Hashing berguna untuk keamanan dan integritas data, serta mempercepat pencarian data.