

Hashovací tabulky

Jiří Zacpal



KATEDRA INFORMATIKY
UNIVERZITA PALACKÉHO V OLOMOUCI

KMI/ZADS - Základní algoritmy a datové struktury



Hashovací funkce

$N=100$

```
def h1(k):  
    return k%N
```

Práce se seznamem

```
def list_insert(l,k):  
    l.append(k)  
  
def list_search(l,k):  
    for i in l:  
        if i==k:  
            return True  
    return False  
  
def list_delete(l,k):  
    if list_search(l,k)==True:  
        l.remove(i)
```

Hashovací tabulka

```
def insert(T, k):  
    i=T["hash"](k)  
    list_insert(T["data"][i], k)  
  
def search(T, k):  
    i = T["hash"](k)  
    return list_search(T["data"][i], k)  
  
def delete(T, k):  
    x=search(T,k)  
    i=T["hash"](k)  
    list_delete(T["data"][i], x)
```



Hashovací funkce pro řetězce

```
import unicodedata
```

```
P=97
```

```
def h2(k):
```

```
    key=unicodedata.normalize('NFKD', k[:3]).encode('ascii', 'ignore')
```

```
    c=key[0]+key[1]*128+key[2]*(128^2)
```

```
    return c%P
```

Otevřené adresování

Otevřené adresování

$N=19$

```
def h1(k):  
    return k%N
```

```
def g1(k,i):  
    return (h1(k) + i)% N
```

Otevřené adresování

```
def insert(T, k):  
    for i in range(len(T["data"])):  
        j=T["g"](k,i)  
        if T["data"][j] == None:  
            T["data"][j]=k  
    return
```

■

Otevřené adresování



```
def search(T,k):  
    for i in range(len(T["data"])):  
        j=T["g"](k,i)  
        if T["data"][j] == k:  
            return k  
        if T["data"][j] == None:  
            break  
    return None
```

Mazání



```
t=[[None,False] for i in range(N)]  
table={"data":t,"g":g1}
```

```
def delete(T,k):  
    for i in range(len(T["data"])):  
        j=T["g"](k,i)  
        if T["data"][j][0] == k:  
            T["data"][j][1] =False  
            return  
        if T["data"][j][0] == None:  
            break  
    return None
```

Kvadratické prozkoumávání

`N=19`

`C1=0`

`C2=1`

`def h1(k):`

`return k%N`

`def g2(k, i):`

`return (h1(k) + C1*i + C2*i^2)%N`

Dvojité hashování

$N=19$

```
def h1(k):  
    return k % N
```

```
def h2(k):  
    return 1 + (k % N - 1)
```

```
def g3(k, i):  
    return (h1(k) + i * h2(k)) % N
```

1. Uložte do hashovací tabulky historická data ze souboru ukol-1.txt (klíčem bude rok a datem popis události).
2. Pro ukládání hodnot do hashovací tabulky použijte metodu dvojitého hashování.
3. Vyhledávací funkce bude vracet popis události (pokud je datum v poli) nebo text „Tento rok se nic nestalo.“

- Příklad:

- Příkazy:

```
print("Rok 650:" + hledej_rok(table,650))  
print("Rok 100:" + hledej_rok(table,100))
```

- Vypíše:

Rok 650:první vlády tyranů

Rok 100:Tento rok se nic nestalo