

Verkefni 9 - Tölvunarfræði 2

ttb3@hi.is

17. mars 2022

1. 3.4.4

skrifaði smá python kóða til að bruteforcea þetta dæmi, prófar fyrstu milljón a fyrir hvert og eitt M, prentar a og M þegar stökin eru 10 mismunandi

```
tolur = [83,69,65,82,67,72,88,77,80,76]

outTolur = []
M = 9
while (True):
    M+=1
    print(M)
    for a in range(1,1000000):
        for k in tolur:
            outTolur.append((a*k)%M)
            if(len(set(outTolur))==10): break
        outTolur = []
    if(len(set(outTolur))==10): break

print('outSet',set(outTolur))
print('outTolur',outTolur)
print('a',a)
print('M',M)
```

a = 1 og M = 20 þá er output [3, 9, 5, 2, 7, 12, 8, 17, 0, 16]

2.

Allar aðgerðir framkvæmdar 10 sinnum

BST

case	insertion time (s)	search time (s)
worst	0.211	0.324
best	0.185	0.218
avg.	0.197	0.274

RedBlackBST

case	insertion time (s)	search time (s)
worst	0.230	0.356
best	0.203	0.221
avg.	0.217	0.254

SCHST

case	insertion time (s)	search time (s)
worst	0.197	0.243
best	0.170	0.219
avg.	0.181	0.227

LPHST

case	insertion time (s)	search time (s)
worst	0.201	0.273
best	0.147	0.219
avg.	0.160	0.236

3. 3.4.31

put

```
...
Key tempKey = null;
Value tempVal = null;

int keyH1 = key.hashCode();

if (keys[0][keyH1] != null) {
    tempKey = keys[0][keyH1];
    tempVal = vals[0][keyH1];
    vals[0][keyH1] = val;
} else {
```

```

        keys[0][keyH1] = key;
        vals[0][keyH1] = val;
    }

    if ((tempKey != null) && (keys[1][tempKey.hashCode()] != null)) {
        vals[1][tempKey.hashCode()] = tempVal;
        Key tTempKey = keys[1][tempKey.hashCode()];
        Value tTempVal = vals[1][tempKey.hashCode()];
        put(tTempKey, tTempVal);
    }
    else {
        keys[1][tempKey.hashCode()] = tempKey;
        vals[1][tempKey.hashCode()] = tempVal;
    }
}

```

get

```

public Value get(Key key) {
    if (key == null)
        throw new IllegalArgumentException("argument to get() is
            null");
    long h = murmurhash(key.hashCode(), 42);
    int h1 = ((int) (h & 0x7fffffff)) % m;
    int h2 = ((int) ((h >> 32) & 0x7fffffff)) % m;
    if (h > h1) return vals[1][(int)h];
    else return vals[0][(int)h];
}

```

4. 4.1.4

Til að finna út hvort vegur liggi á milli tveggja hnúta í óstefndu neti þarf bara að athuga hvort það sé tenging á milli þeirra. Graph hefur poka af pokum sem halda utan um tengingar hvers og eins hnútar. Þar sem þessi gögn eru auðveldlega sótt er málið einfaldlega að tékka á hvort hnútur V sem er tekinn inn hafi hnút W í poka sínum.

```

public boolean hasEdge(int V, int W) {
    for (int node : adj(v)) {
        if (node.equals(w)) return true;
    }
    return false;
}

```
