Exercises 8

张逸松

November 17, 2019

9.19

```
template <typename Key, typename E> bool hashdict<Key, E> :: heshDelete(
   const Key &k, const E &e) const {
    int init;
    int pos = init = h(k);
    for (int i = 1; k != HT[pos].key() && EMPTYKEY != HT[pos],key(); i
   ++) pos = (init + p(k, i)) % M;
    if (EMPTYKEY == HT[pos].key()) return false;
    e = HT[pos];
    HT[pos] = TOMBSTONE;
    return true;
}
template <typename Key, typename E> bool hashdict<Key, E> :: heshInsert(
   const Key &k, const E &e) {
    int init;
    int pos = init = h(k);
    for (int i = 1; EMPTYKEY != HT[pos].key() && TOMBSTONE != HT[pos].
   key(); i++) {
        pos = (init + p(k, i)) % M;
        Assert(k != HT[pos].key(), "Duplicates not allowed");
    }
    KVpair <Key, E> temp(k, e);
    HT[pos] = temp;
    return true;
}
template <typename Key, typename E> E hashdict<Key, E> :: hashSearch(
   const Key &k) const {
    int init;
    int pos = init = h(k);
    for (int i = 1; k != HT[pos].key() && EMPTYKEY != HT[pos].key(); i
   ++) pos = (init + p(k, i)) % M;
    if (k == HT[pos].key()) return HT[pos].value();
    return NULL;
}
```

10.9

M -> D I M -> S

D I -> B D I -> G D I -> K

S -> P S -> U

B -> A B -> C

G -> E G -> H

K -> J K -> L

P -> N O P -> R

U -> T U -> W

10.12

18|33 -> 4|10 18|33 -> 23 18|33 -> 48

4|10 -> |1 2 3| 4|10 -> |4 5 6| 4|10 -> |10 12 15|

23 -> |18 19 20 21 22| 23 -> |23 30 31|

48 -> |33 45 47| 48 -> |48 50 52|

10.13

23|33 -> 10|12|15|21|22 23|33 -> 23|30|31 23|33 -> 45|47|48|50|52