

Hausaufgabe 7.1: Lineares Hashing mit partiellen Erweiterungen

a) ...

b) *Algorithmus* :

```
boolean suche(int key)
{
    // Adresse für noch nicht gesplittete Seiten berechnen
    int adresse = getHashValue(key, expansionsLevel, aktPartielleExpansion);

    // Stelle fest, ob schon gesplittet
    if ((adresse%(Math.pow(2,expansionsLevel)*(startGroesse/2))) < expansionsZeiger)
    {
        // Falls bereits gesplittet, verwende naechste Hashfunktion
        adresse = getHashValue(key, expansionsLevel, aktPartielleExpansion+1);
    }
    // Durchsuche Seite mit berechneter Adresse
    HashSeite zuPruefendeSeite = (HashSeite) primaerSeiten.elementAt(adresse);

    return zuPruefendeSeite.durchsuche(key);
}
```

c) *Algorithmus*

```

void expandiereHashtabelle()
{
    // Einfuegen einer neuen Primaerseite
    primaerSeiten.add(new HashSeite());

    // Durchlaufe alle Seiten die durch den Expansionszeiger gesplittet werden sollen
    for (int j=0; j<= aktPartielleExpansion; j++)
    {
        int neuEinZuSortieren[];
        HashSeite zuUeberpruefendeSeite;

        // Berechne Adresse in Abhaengigkeit von j
        int adresseZuUebSei = expansionsZeiger
            +j*((int)(Math.pow(2,expansionsLevel)))*(startGroesse/2);

        // Hole alle Schluessel der Seite
        zuUeberpruefendeSeite = (HashSeite) primaerSeiten.elementAt(adresseZuUebSei);
        neuEinZuSortieren = zuUeberpruefendeSeite.alleSchluessel();

        // Durchlaufe gefundene Schluessel und entscheide, ob umgespeichert werden muss
        for (i=0; i<neuEinZuSortieren.length; i++)
        {
            int aktkey = neuEinZuSortieren[i];
            // Berechne Hashvalue nach alter Funktion
            int hashValueAlt = getHashValue(aktkey, expansionsLevel, aktPartielleExpansion);

            // Berechne Hashvalue nach neuer Funktion
            int hashValueNeu = getHashValue(aktkey, expansionsLevel, aktPartielleExpansion+1);

            if (hashValueAlt != hashValueNeu)
            {
                zuUeberpruefendeSeite.loesche(aktkey);
                ((HashSeite) primaerSeiten.elementAt(hashValueNeu)).fuegeEin(aktkey);
            }
        }
    }

    // Inkrementiere ExpansionZeiger, partielle Expansion und Expansionslevel
    expansionsZeiger = expansionsZeiger + 1;

    // Falls groesser als Startgroesse * 2^(L-1)
    if (expansionsZeiger == Math.pow(2,expansionsLevel)*(startGroesse/2))
    {
        expansionsZeiger = 0;

        // inc partielle Exp
        aktPartielleExpansion = aktPartielleExpansion + 1;

        // inc Level
        if (aktPartielleExpansion > anzahlPE)
        {
            aktPartielleExpansion = 1;
            expansionsLevel = expansionsLevel + 1;
        }
    }
}

```