

SEMINAR 6 – TABELAT HASH

1. Shtoni çelësat E A S Y Q U T I O N në mënyrë sekuenciale në një tabelë fillimisht boshe, ku $M=5$ lista, dhe metoda e përdorur është separate chaining. Përdorni funksionin $hash(I) = 11 * k \% M$ për të transformuar shkronjën e k -të të alfabetit në një index në tabelë.

Shembull: $hash(I) = hash(9) = 99 \% 5 = 4$

A	B	C	D	E	F	G	H	I	J
1	2	3	4	5	6	7	8	9	10
K	L	M	N	O	P	Q	R	S	T
11	12	13	14	15	16	17	18	19	20
U	V	W	X	Y	Z				
21	22	23	24	25	26				

2. Write a program to find values of a and M , with M as small as possible, such that the hash function $(a * k) \% M$ for transforming the k th letter of the alphabet into a table index produces distinct values (no collisions) for the keys S E A R C H X M P L. The result is known as a *perfect hash function*.
3. A është implementimi i mëposhtëm i lejuar për një funksion hash? Argumentoni.

```
public int hashCode()
{
    return 17;
}
```

4. Tregoni përmbajtjen e një table hash me linear-probing, që krijohet si rezultat i shtimit sekuencial të çelësve: E A S Y Q U T I O N në një tabelë fillimisht boshe me përmasë $M=4$. Kjo tabelë dyfishohet sa herë është e mbushur përgjysëm. Përdorni funksionin $11 * k \% M$ për të transformuar shkronjën e k -të të alfabetit në një indeks në tabelën hash.
5. Which of the following scenarios leads to expected *linear* running time for a random search hit in a linear-probing hash table?
 - a. All keys hash to the same index.
 - b. All keys hash to different indices.
 - c. All keys hash to the same even-numbered index.
 - d. All keys hash to different even-numbered indices.

6. Suppose that the keys A through G, with the hash values given below, are inserted in some order into an initially empty table of size 7 using a linear-probing table (with no resizing for this problem).

key	A	B	C	D	E	F	G
hash (M = 7)	2	0	0	4	4	4	2

Which of the following could not possibly result from inserting these keys?

- a. E F G A C B D
- b. C E B G F D A
- c. B D F A C E G
- d. C G B A D E F
- e. F G B D A C E
- f. G E C A D B F

Give the minimum and the maximum number of probes that could be required to build a table of size 7 with these keys, and an insertion order that justifies your answer.

7. How many compares could it take, in the worst case, to insert N keys into an initially empty table, using linear probing with array resizing? Consider an initial table size of N .
8. Bad hash function. Consider the following hashCode() implementation for String, which was used in early versions of Java:

```
public int hashCode()
{
    int hash = 0;
    int skip = Math.max(1, length()/8);
    for (int i = 0; i < length(); i += skip)
        hash = (hash * 37) + charAt(i);
    return hash;
}
```

Explain why you think the designers chose this implementation and then why you think it was abandoned in favor of the following code:

```
public int hashCode()
{
    int hash = 0;
    for (int i = 0; i < length(); i++)
        hash = (hash * 31) + charAt(i);
    return hash;
}
```