

hashfun → ASCII

A → 45 → 65 % 4 = 1  
B → 68 → 66 % 4 = 2  
C → 97 → 67 % 4 = 3  
d → 48 → 68 % 4 = 0  
E → 77 → 69 % 4 = 1



n = ✓

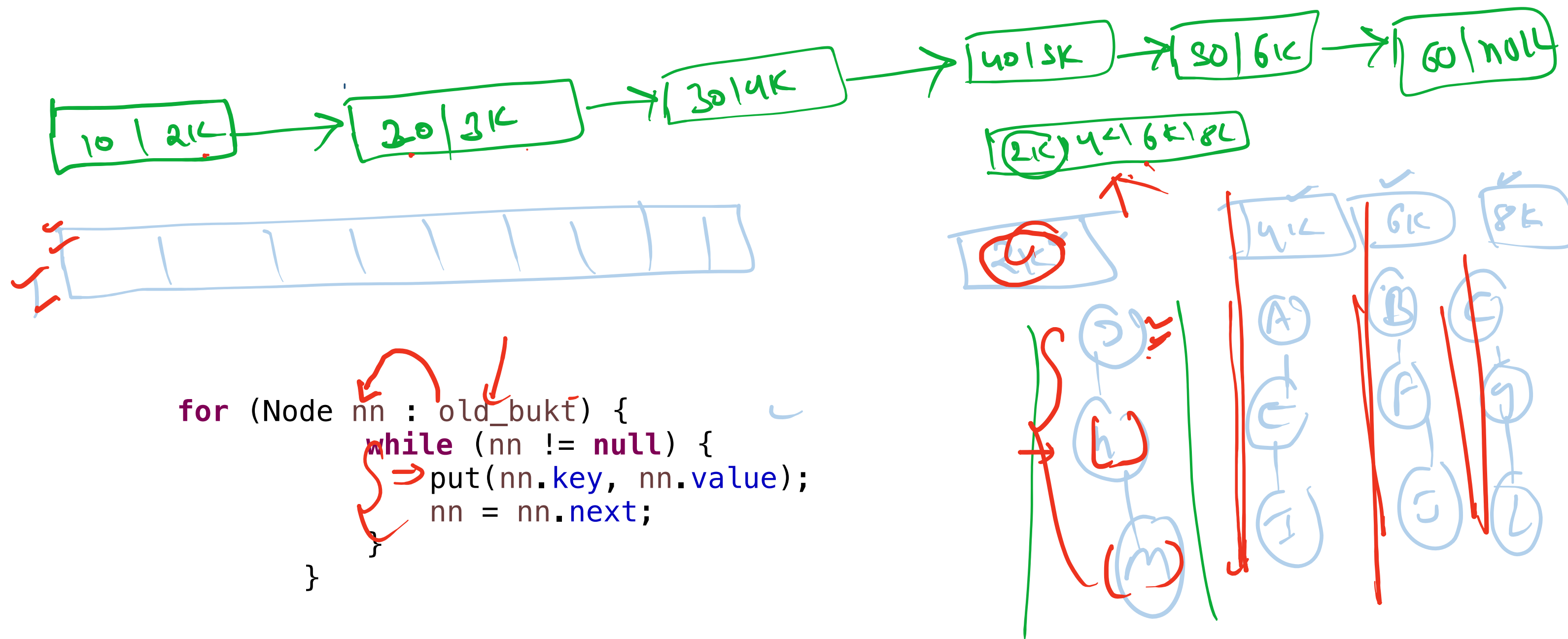
THF = 2.0

1f = 12/4 = 3

12/8 = 1.5

axp + bxp + ...

```
public void put(K key, V value) {  
    int bn = hashfun(key);  
    Node temp = bukt.get(bn);  
    while (temp != null) {  
        if (temp.key == key) {  
            temp.value = value;  
        }  
        temp = temp.next;  
    }  
    Node nn = new Node();  
    nn.key = key;  
    nn.value = value;  
    size++;  
    temp = bukt.get(bn);  
    nn.next = temp;  
    bukt.set(bn, nn);  
}
```



```
public String toString() {  
    String s = "{";  
    for (Node nn : bukt) {  
        while (nn != null) {  
            s = s + nn.key + "=" + nn.value + "  
            nn = nn.next;  
        }  
    }  
    return s + "}";  
}
```

ac  
97 59  
bb  
98 98

Input: strs = ["eat", "tea", "tan", "ate", "nat", "bat"]

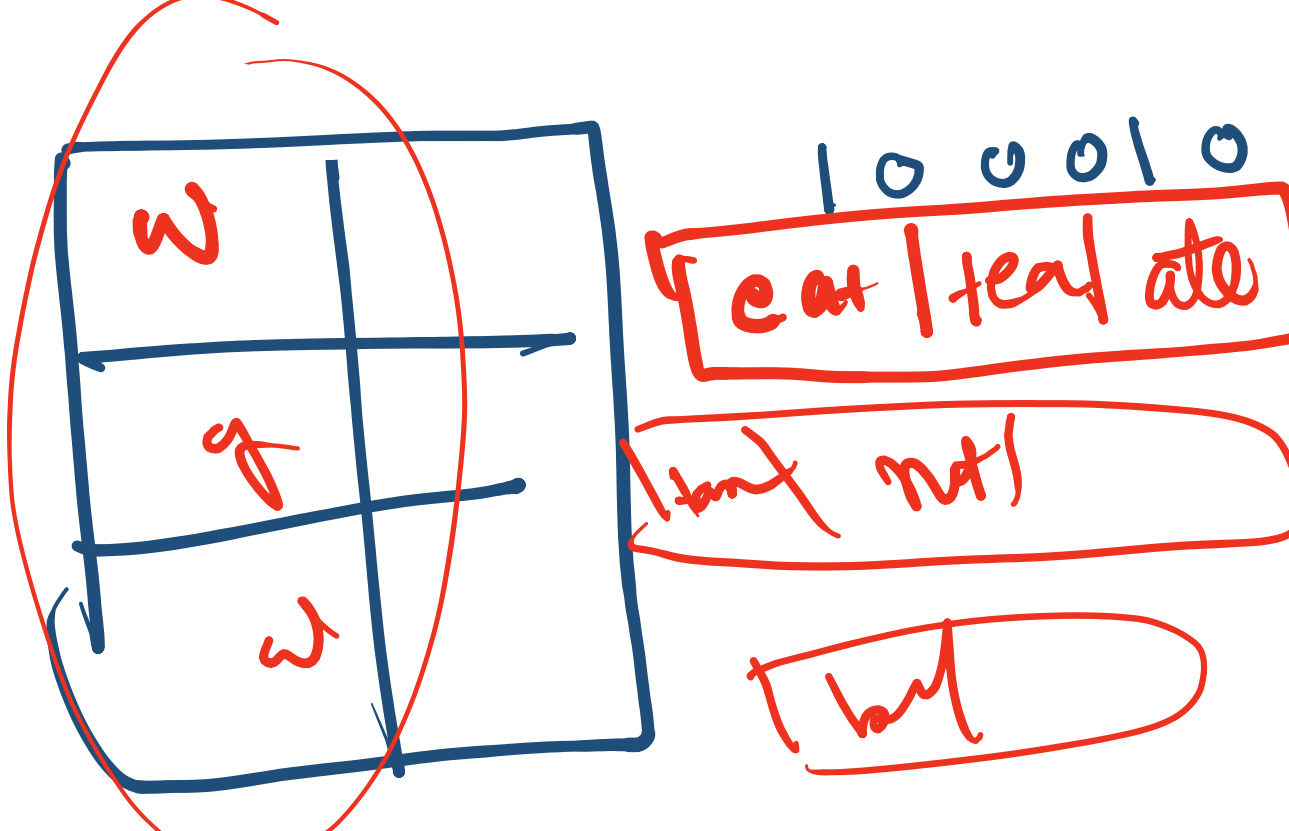
Output: [ ["bat"], ["nat", "tan"], ["ate", "eat", "tea"] ]

Explanation:

- There is no string in strs that can be rearranged to form "bat".
- The strings "nat" and "tan" are anagrams as they can be rearranged to form each other.
- The strings "ate", "eat", and "tea" are anagrams as they can be rearranged to form each other.

w	2k	eat tea ate
y	3k	tan tan
z	4k	bat

```
HashMap<String, List<String>> map = new HashMap<>();  
for (int i = 0; i < arr.length; i++) {  
    String key = Anagram_Key(arr[i]);  
    if (!map.containsKey(key)) {  
        map.put(key, new ArrayList<>());  
    }  
    map.get(key).add(arr[i]);  
}
```



Wrong Answer Details >

Input	["bddddddddd", "bbbbbbbbbbbc"]
Output	["bddddddddd", "bbbbbbbbbbbc"]
Expected	["bbbbbbbbbbbc", "bddddddddd"]

010100  
010100  
010100  
010100