Problem 3:

1. What test files did you use (describe them)?
   First Test: the entire text of Hamlet as suggested
   Second test: 1000 randomly generated chars (capital letters, lowercase letters, numbers)
   Third test: a paper I wrote for another class (about 1300 words)

2. What was the capacity of your cache for each test?
   Hamlet: 1000
   Rand Chars: 20 (because only 62 total different chars)
   Paper: 200

3. What was the total number of rotations for each file?
   Hamlet: 170417 left, 166405 right (336822 total)
   Rand Chars: 3673 left, 3567 right (7240 total)
   Paper: 7504 left, 7476 right (14980 total)

4. What was the size of each file?
   Hamlet File: 32058 words
   Random Char File: 1000 chars
   Paper File: 1333 words

5. What was the average number of rotations per item (that is, your answer to 3, divided by your answer to 4)?
   Hamlet: 10.5 (10-11 average rotations)
   Rand Chars: 7.24
   Paper: 11.23

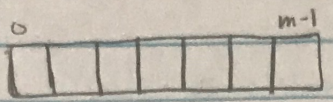
6. How many items did you have to remove from the cache? This occurs when you bring a new item into the cache, which is referred to as a `cache miss`. Caches are designed in such a way as to minimize cache misses.
   Hamlet: 22277
   Rand Chars: 633
   Paper: 598

7. Was there a noticeable difference between the two moderate-sized tests? Explain why there was or was not a difference.

The peper file had many more rotations than the random chars file. Since the chars file only had 62 different characters, and the paper file has many more different words, the chars file will have to insert new keys and splay from the bottom much less frequently, because the likelihood that the key is already in the splay tree and has been recently used is greater.

8. Include any other interesting analysis your test cases revealed.
   if you decrease the cache, the num of rotations goes down, because each time something new is inserted there are not as many rotations needed to splay the node

# Problem 2

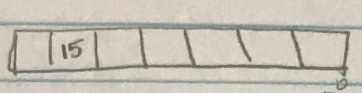



hashtable $m = 7$

- maintain load factor $\leq 1/2$ (if $> 1/2$, resize to 11)

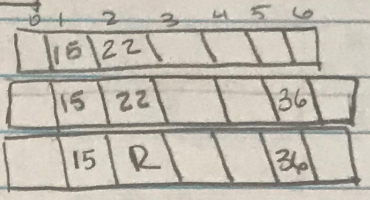
Part a) use quadratic probing

insert 15: $h(15) = 15\%7 = 1$




insert 22: $h(22) = 22\%7 = 1 \Rightarrow$ insert at $1+1^2 = 2$

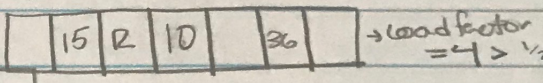
insert 36: $h(36) = 36\%7 = 1 \Rightarrow$ insert at $1+2^2 = 5$



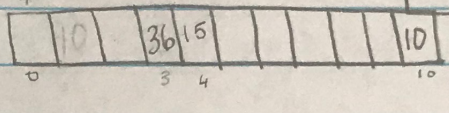

remove 22:

Find 36: searches location 1, then 2, then 5 → returns true

insert 10: $h(10) = 10\%7 = 3$

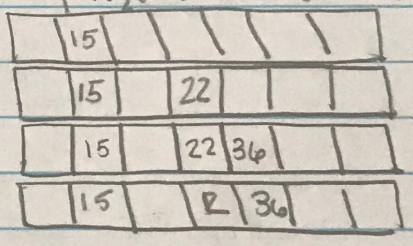



→ resize: (+rehash)




b) use double hashing $h_1(k) = k\%m$, $h_2(k) = 3-(k\%3)$

insert 15: $h_1(15) = 1$



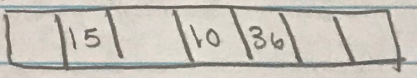

insert 22: $1 + (3-(22\%3)) = 3$

insert 36: $1 + (3-(36\%3)) = 4$

remove 22:

find 36: searches location 1, then 4 → returns true

insert 10: $h_1(10) = 3$




c) $P(h_i(x) = p) = \frac{1}{m}$

Prob(false positive) $= (1-\frac{m}{3})^3$ ↳ prob false ↳ 3 hash fns

$$\sum_{i=1}^{27} (1-\frac{m}{3})^3 = 27(1-\frac{m}{3})^3$$