

1.

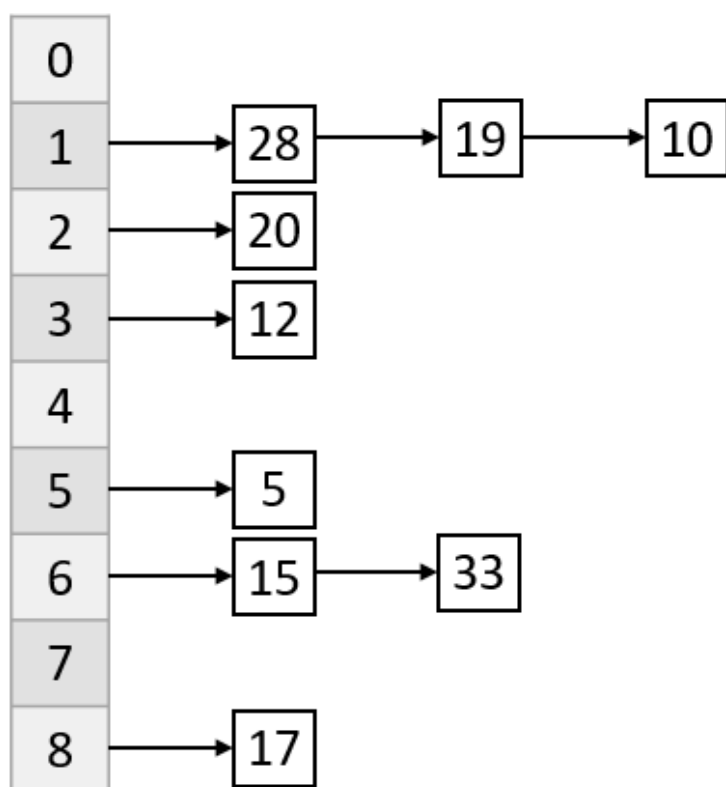
- 根據題目所述，需要traverse的次數比insert的次數還要多，因此會優先以四種實作方式各自的retrieval時間複雜度為首要考量，若相同，再比較insertion的時間複雜度：

- 四種實作方式的 retrieval & insertion 時間複雜度：

	Retrieval	Insertion
sorted array based	$O(\log n)$	$O(n)$
sorted linked based	$O(n)$	$O(n)$
binary search tree	$O(\log n)$	$O(\log n)$
hashing	$O(1) \sim O(n)$	$O(1)$

- 綜合Retrieval和Insertion的時間複雜度，binary search tree和hashing都有較好的表現，不過要怎麼從中抉擇呢？若滿足以下條件，則選擇hashing：

01. 能事先能知道dictionary的size：當hash table太小時會頻繁發生collision，最糟情況下會到達worst case— $O(n)$ ，所以在使用hashing前必須先能得知資料數量，否則就使用binary search tree。
02. 所有資料是無序的：在hash table中的所有資料是沒有經過排序的，而binary search tree有，當我們對資料有排序的需要時就選擇binary search tree。以此題的spell checker所儲存的字典來說是不需要的，所以我們選擇hashing。



$$m = 1000$$

$$A = ((5^{1/2} - 1) / 2) \text{ 大約等於 } 0.6180339887498949$$

$$h(k) = [m (kA \bmod 1)]$$

$$h(61) = 700$$

$$h(62) = 318$$

$$h(63) = 936$$

$$h(64) = 554$$

$$h(65) = 172$$

linear probing

0		0	22	0	22	0	22	0	22	0	22
1		1		1		1		1		1	
2		2		2		2		2		2	
3		3		3		3		3		3	
4		4		4		4	4	4	4	4	4
5		5		5		5		5	15	5	15
6		6		6		6		6		6	28
7		7		7		7		7		7	
8		8		8		8		8		8	
9		9		9	31	9	31	9	31	9	31
10	10	10	10	10	10	10	10	10	10	10	10

0	22	0	22	0	22
1		1	88	1	88
2		2		2	
3		3		3	
4	4	4	4	4	4
5	15	5	15	5	15
6	28	6	28	6	28
7	17	7	17	7	17
8		8		8	59
9	31	9	31	9	31
10	10	10	10	10	10

quadratic probing

0	
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	10

0	22
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	10

0	22
1	
2	
3	
4	
5	
6	
7	
8	
9	31
10	10

0	22
1	
2	
3	
4	4
5	
6	
7	
8	
9	31
10	10

0	22
1	
2	
3	
4	4
5	15
6	
7	
8	
9	31
10	10

0	22
1	
2	
3	
4	4
5	15
6	28
7	
8	
9	31
10	10

0	22
1	
2	
3	
4	4
5	15
6	28
7	17
8	
9	31
10	10

0	22
1	88
2	
3	
4	4
5	15
6	28
7	17
8	
9	31
10	10

0	22
1	88
2	
3	
4	4
5	15
6	28
7	17
8	59
9	31
10	10

double hashing

0	
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	10

0	22
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	10

0	22
1	
2	
3	
4	
5	
6	
7	
8	
9	31
10	10

0	22
1	
2	
3	
4	4
5	
6	
7	
8	
9	31
10	10

0	22
1	
2	
3	
4	4
5	15
6	
7	
8	
9	31
10	10

0	22
1	
2	
3	
4	4
5	15
6	28
7	
8	
9	31
10	10

0	22
1	
2	
3	17
4	4
5	15
6	28
7	
8	
9	31
10	10

0	22
1	
2	
3	17
4	4
5	15
6	28
7	88
8	
9	31
10	10

0	22
1	
2	59
3	17
4	4
5	15
6	28
7	88
8	
9	31
10	10