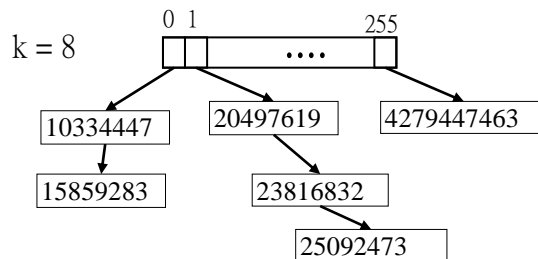


C Language Programming: Homework #8
Assigned on 01/01/2017(Sunday), Due on 01/16/2017(Monday)

1. This homework ask you to do the search, insert, and delete operations of 32-bit undigned numbers generated randomly based on singly linked list.
2. In addition to the linked lists, you have to use an array called segmentation tables of size 2^k , where k is an input from the command line and construct the following data structure:



The numbers in the linked list are in the increasing order. Take $k = 8$ as an example and there are 6 numbers. There are 256 segments (segment 0 to segment 255). If the most significant 8 bits of a number is equal to i , this number will be inserted into the linked list of segment i . For example, the most significant 8 bits of numbers 10334447 and 15859283 is 0 and so they are inserted into the linked list of segment 0.

3. generate 32-bit unsigned numbers by using two random numbers from `rand()` as follows:
`a=rand(); a=(a & 1) << 31; r=rand(); r = r | a;`
4. First generate $n=100000$ numbers and keep record the first $m=1000$ numbers and insert these 100000 numbers into the data structure. Then search the recorded 1000 numbers and measure the average search time.
5. Then generate another 1000 numbers and insert them into the data structure and measure the average insert time.
6. For deletion, delete the above 1000 numbers and measure the average the delete time.
7. Please select some different k , n , and m to get these average times.
8. Use the following code to measure the time:

```
// gcc -c -g -Wa,-a,-ad rdtsc-test.c > rdtsc-test.lst
#include <stdio.h>
unsigned long long begin,end, total_time=0;
inline unsigned long long int rdtsc()
{ unsigned long long int x;
  asm volatile ("rdtsc" : "=A" (x));
  return x;}
```

```
inline unsigned long long int rdtsc_32bits()
{ unsigned long long int x;
  asm volatile ("rdtsc" : "=A" (x));
  return x;}
```

```
inline unsigned long long int rdtsc_64bits()
{ unsigned long long int x;
  unsigned a, d;
```

```
__asm__ volatile("rdtsc" : "=a" (a), "=d" (d));
```

```
return ((unsigned long long)a | (((unsigned long long)d) << 32));}
```

```
int main(int argc, char *argv[])
{
    unsigned i, j, x = 105; long li, cnt=100;
    unsigned long b0,b1,e0,e1; unsigned long long b, c, e, f;

    if (argc == 2) cnt=atoi(argv[1]);
    asm __volatile__ ( " rdtsc  " : "=a" (b));
    printf("b = %u\n", b);
    for(j=0; j<10; j++){    i += j; }
    asm __volatile__ ( " rdtsc  " : "=a" (c));

    printf("c = %d\n", c); printf("c-b = %d\n", c-b);

    li=0;
    begin=rdtsc();
    for(j=0; j<cnt; j++){    li += j; }
    end=rdtsc();
    total_time=end-begin;
    printf("%lu clock cycles (rdtsc_32bits)\n",total_time);

    li=0;
    begin=rdtsc_64bits();
    for(j=0; j<cnt; j++){    li += j; }
    end=rdtsc_64bits();
    total_time=end-begin;
    printf("%lu clock cycles (rdtsc_64bits)\n",total_time);
}
```