\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Report: Author: F74055047張財實 <changchaishi@gmail.com>

Class: 乙班

Description:

靈活運用link list！

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Code:

#include <stdio.h>

#include <math.h>

#include <stdlib.h>

#include <time.h>

int k,m,n;//global variable

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

自造亂數

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

unsigned int rand\_num(){

unsigned int a,r;

a=rand(); a=(a & 1) << 31; r=rand(); r = r | a;

return r;

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

時間計算

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

inline unsigned long long int rdtsc(){

unsigned long long int x;

asm volatile ("rdtsc":"=A"(x));

return x;

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Link list 宣告

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

typedef struct node \*node\_ptr;

typedef struct node{

unsigned int data;

node\_ptr next;

}node;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

基本insert /search /delete a node的函式

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

node \*insert\_a\_node(node \*head, node \*p)

{

node \*t=head;

if(p==NULL) return head;

if(head == NULL) return p;

if (p->data <= head->data){

p->next = head;

return p;

}

while (t->next !=NULL && t->next->data < p->data)

t = t->next;

p->next = t->next;

t->next = p;

return head;

}

node \*search\_a\_node(node \*head, int d)

{

node \*t=head;

while (t != NULL && t->data != d){

t = t -> next;

}

return t;

}

node \*delete\_a\_node(node\* head,unsigned int d)

{

node\* cur,\*prev;

for(cur=head,prev=NULL;cur!=NULL && cur->data!=d;prev=cur,cur=cur->next);

if(cur==NULL)

return head;

if(prev==NULL)

head=head->next;

else

prev->next=cur->next;

free(cur);

return head;

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Empty node generator

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

node\_ptr new\_node(){

node\_ptr t;

if (!(t=malloc(sizeof(\*t))))exit(EXIT\_FAILURE);

t->next=NULL;

t->data=0;

return t;

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

運算該亂數的table位置的index

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

int table\_num(unsigned int num){

int i,j,rank=0;

for(i=31,j=k-1;i>=31-k,j>=0;i--,j--)

if(num&1<<i)

rank|=1<<j;

return rank;

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Data base +整合首m個亂數

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void create\_list(node\_ptr table[],unsigned int num1[]){

int i;

unsigned int num,rank;

node\_ptr add;

for(i=0;i<n;i++){

num=rand\_num();

if(i<m)num1[i]=num;

rank=table\_num(num);

add = new\_node();

add->next=NULL;

add->data=num;

table[rank]->next=insert\_a\_node(table[rank]->next,add);

}

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Search /insert /delete list的函式

void search\_list(node\_ptr table[],unsigned int num1[]){

node\_ptr temp;

int j;

unsigned int rank;

FILE \*fp;

fp=fopen("result.txt","w");

for(j=0;j<m;j++){

rank=table\_num(num1[j]);

temp=search\_a\_node(table[rank]->next,num1[j]);

fprintf(fp, "%u\n", temp->data);

}

fclose(fp);

}

void insert\_list(node\_ptr table[],unsigned int num1[]){

int i;

unsigned int num,rank;

node\_ptr add;

for(i=0;i<m;i++){

num1[i]=rand\_num();

rank=table\_num(num1[i]);

add=new\_node();

add->next=NULL;

add->data=num1[i];

table[rank]->next=insert\_a\_node(table[rank]->next,add);

}

}

void delete\_list(node\_ptr table[],unsigned int num1[]){

int i;

unsigned int num,rank;

for(i=0;i<m;i++){

rank=table\_num(num1[i]);

table[rank]->next=delete\_a\_node(table[rank]->next,num1[i]);

}

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Main bj4

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

int main(int argc, char \*argv[]){

srand((unsigned) time(NULL));

int i;

k=atoi(argv[1]);

n=atoi(argv[2]);

m=atoi(argv[3]);

unsigned int array\_num,num1[m];

array\_num=pow(2,k);

node\_ptr table[array\_num];

for(i=0;i<array\_num;i++)

table[i]=new\_node();

create\_list(table,num1);

unsigned long long int begin,end,result;

for(i=1;i<=3;i++){

printf("qq i = %d\n", i);

begin=rdtsc();

switch(i){

case 1:search\_list(table,num1);break;

case 2:insert\_list(table,num1);break;

case 3:delete\_list(table,num1);break;

}

end=rdtsc();

result=end-begin;

printf("%lld\n",(result/m));

}

return 0;

}

Compilation:

gcc –o hw8 hw8.c -lm

Execution:

./hw8 2 5 2

Output(program):

275973

2916

3373

Output(result.txt):

2987059560

587228684