// Ding Zhu

// CSC382 2/28/17

#include <iostream>

#include <stdio.h>

#include <time.h>

using namespace std;

int Find(int x, int A[], int n) {//array of sizen

int j;

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

if (x == A[j])

return (j + 1);

}

return 0;

}

int sequence(bool real, int n, int repeat, int bound) {

int \*A = new int[n];

int hits = 0;

int totalSteps= 0;

for (int k = 0; k < repeat; k++) {

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

A[i] = rand() % bound;

}

int x = rand() % bound;

if (real == true) {

int z = Find(x, A, n);

if (z == 0)

totalSteps += 50;

else

totalSteps += z;

}

if (Find(x, A, n) != 0 && real != true)

hits++;

}

if (real == true) {

return totalSteps;

}

else

return hits;

}

void main() {

int n = 50;

int hits = 0;

int bound = 0;

int repeat = 10000;

int steps;

float cAverage, rAverage;

srand(time(NULL));

printf("%s%25s%25s", "Bound", "Calculated Average", "Real Average\n");

for (bound = 30; bound <= 10000; bound+=10) {

if (bound == 30 || bound == 80 || bound == 100 || bound == 1000 || bound == 10000) {

float q = (float)sequence(false, n, repeat, bound)/10000;

steps = sequence(true, n, repeat, bound);

cAverage = n + (q / 2) - ((q\*n) / 2);

rAverage = (float)steps / repeat;

printf("%5d%25.3f%25.3f\n", bound, cAverage, rAverage);

}

}

system("pause");

}

