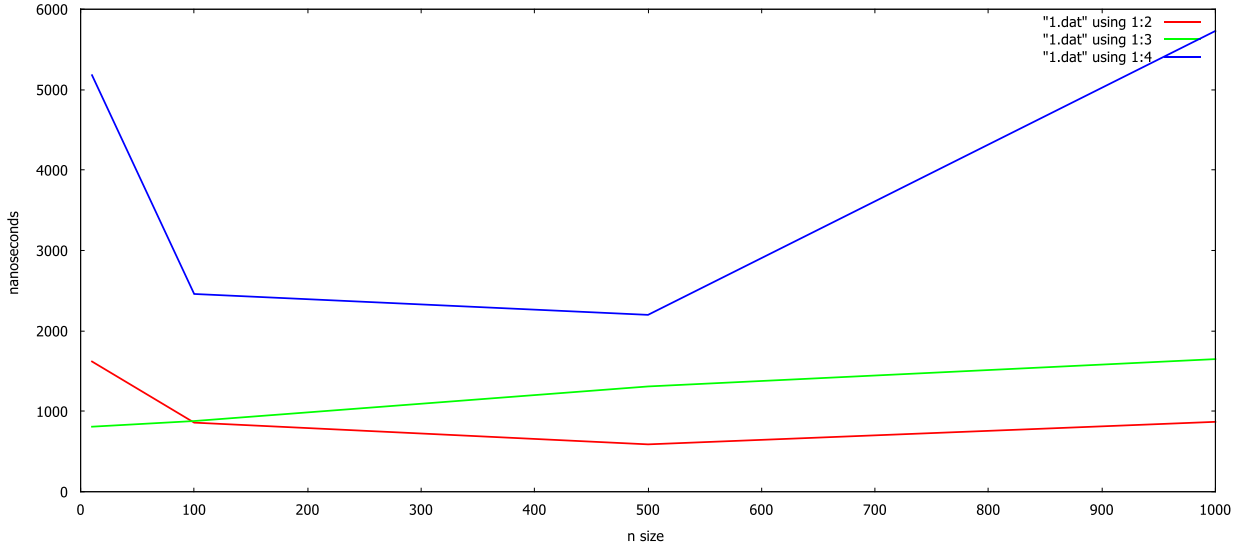
I have to use the data run my program from https://www.onlinegdb.com/online_c++_compiler.

My computer seems too fast, I get all zero at runtime, even use nanoseconds.

I will show printscreen of www.onlinegdb.com/online_c++_compiler at final



Red Line – randomized-select

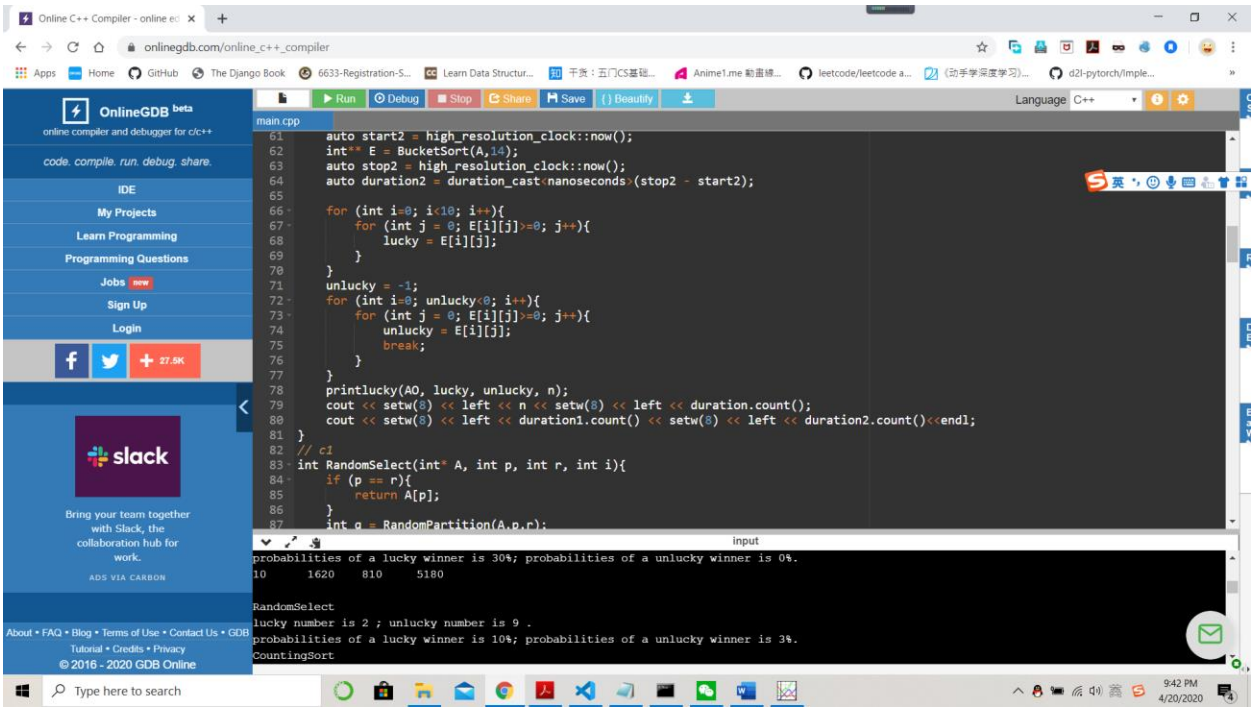
Green Line – counting sort

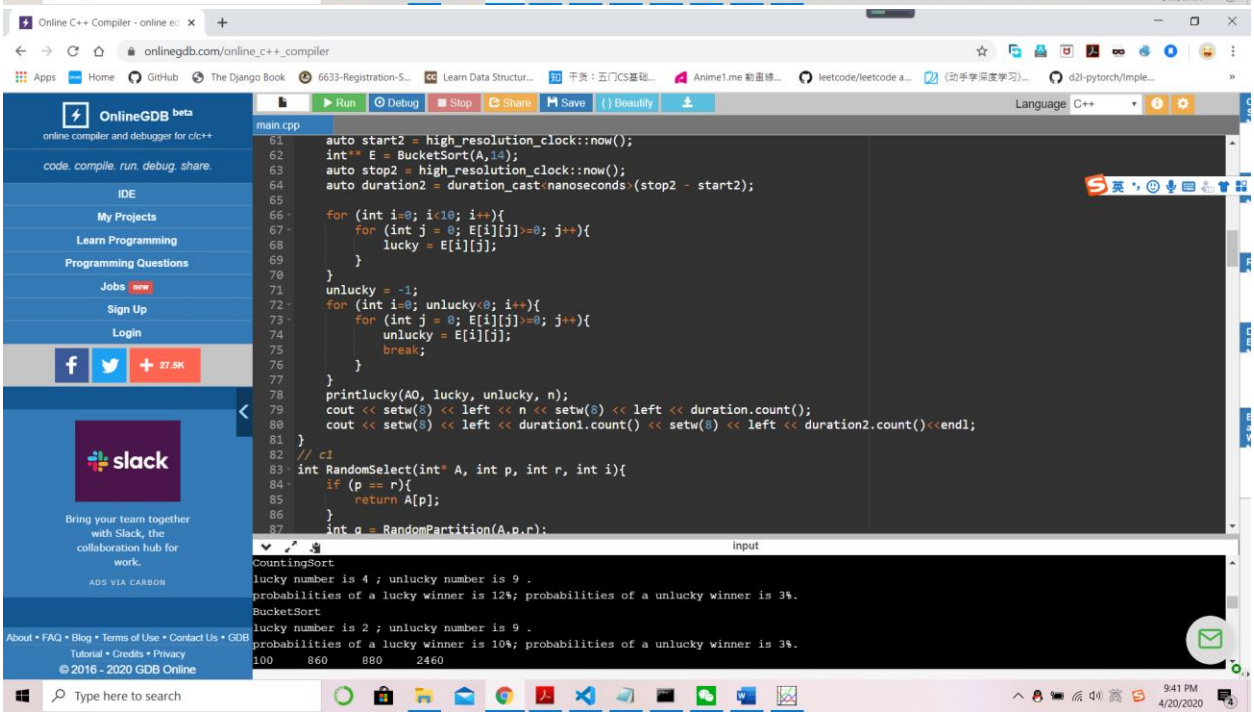
Blue Line – bucket sort

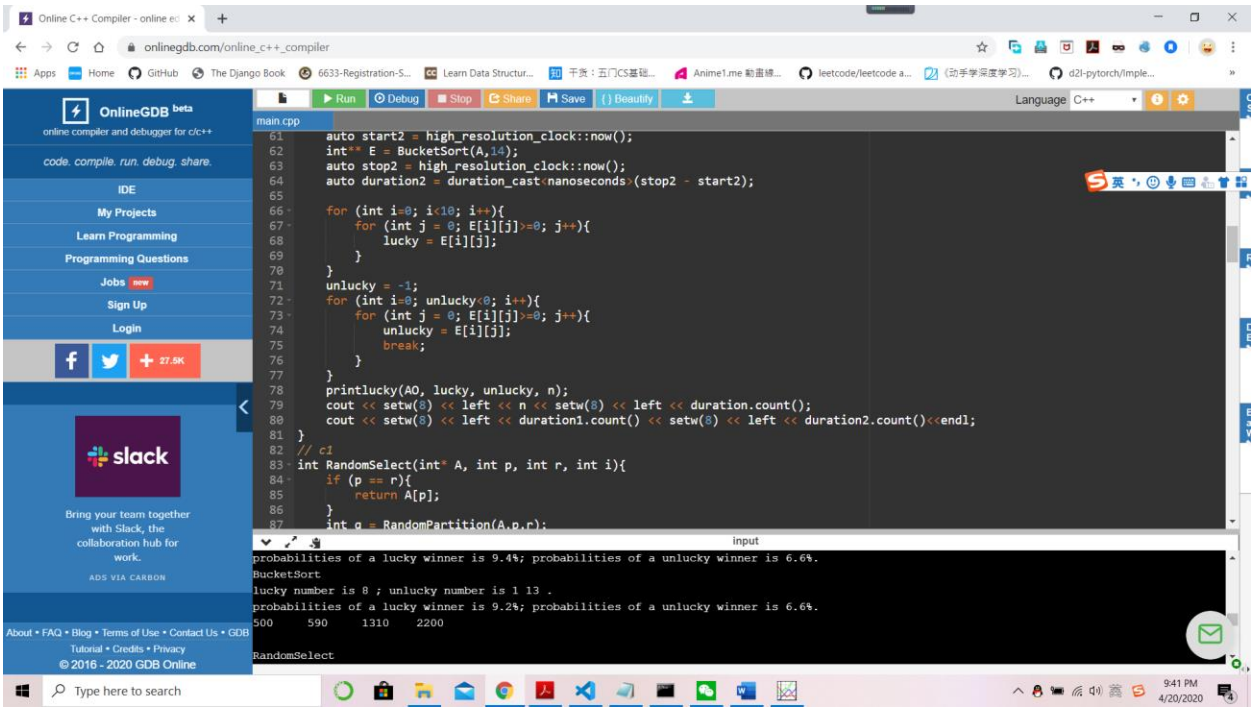
Explanation:

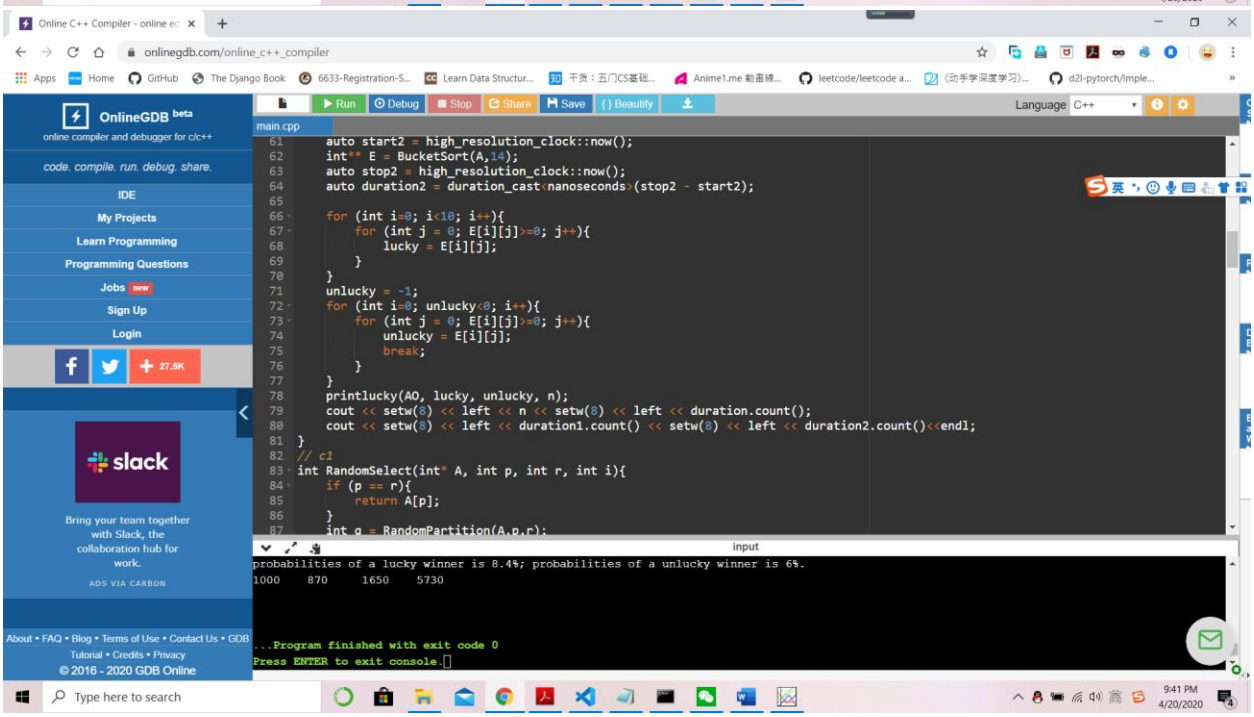
1. Time complexity of counting sort is O(n), when data closer, time complexity of randomized-select will be closer to O(n). When n increase, the randomness decrease, which make data closer
2. The reason why runtime of bucket sort is unstable may be that I use insertion sort to sort bucket as book written.
3. Probabilities of lucky and unlucky is closer when n grow higher, because as n increases, the randomness decreases

```
BUCKET-SORT(A)
1  let B[0..n − 1] be a new array
2  n = A.length
3  for i = 0 to n − 1
4      make B[i] an empty list
5  for i = 1 to n
6      insert A[i] into list B[⌊nA[i]⌋]
7  for i = 0 to n − 1
8      sort list B[i] with insertion sort
9  concatenate the lists B[0], B[1], ..., B[n − 1] together in order
```


Online C++ Compiler - online ed
onlinegdb.com/online_c++_compiler
OnlineGDB beta
online compiler and debugger for c/c++
code. compile. run. debug. share.
IDE
My Projects
Learn Programming
Programming Questions
Jobs new
Sign Up
Login
27.5K
slack
Bring your team together with Slack, the collaboration hub for work.
ADS VIA CARBON
About • FAQ • Blog • Terms of Use • Contact Us • GDB Tutorial • Credits • Privacy
© 2016 - 2020 GDB Online
Run
Debug
Stop
Share
Save
{ } Beautify
Language C++
main.cpp
auto start2 = high_resolution_clock::now();
int** E = BucketSort(A,14);
auto stop2 = high_resolution_clock::now();
auto duration2 = duration_cast<nanoseconds>(stop2 - start2);
for (int i=0; i<10; i++){
for (int j = 0; E[i][j]>=0; j++){
lucky = E[i][j];
}
}
unlucky = -1;
for (int i=0; unlucky<0; i++){
for (int j = 0; E[i][j]>=0; j++){
unlucky = E[i][j];
break;
}
}
printlucky(AO, lucky, unlucky, n);
cout << setw(8) << left << n << setw(8) << left << duration.count();
cout << setw(8) << left << duration1.count() << setw(8) << left << duration2.count()<<endl;
}
// c1
int RandomSelect(int* A, int p, int r, int i){
if (p == r){
return A[p];
}
int q = RandomPartition(A,p,r);
input
probabilities of a lucky winner is 30%; probabilities of a unlucky winner is 0%.
10 1620 810 5180
RandomSelect
lucky number is 2 ; unlucky number is 9 .
probabilities of a lucky winner is 10%; probabilities of a unlucky winner is 3%.
CountingSort
Type here to search
9:42 PM
4/20/2020


Online C++ Compiler - online ed
onlinegdb.com/online_c++_compiler
OnlineGDB beta
online compiler and debugger for c/c++
code. compile. run. debug. share.
IDE
My Projects
Learn Programming
Programming Questions
Jobs new
Sign Up
Login
27.5K
slack
Bring your team together with Slack, the collaboration hub for work.
ADS VIA CARBON
About • FAQ • Blog • Terms of Use • Contact Us • GDB Tutorial • Credits • Privacy
© 2016 - 2020 GDB Online
Run
Debug
Stop
Share
Save
{ } Beautify
Language C++
main.cpp
auto start2 = high_resolution_clock::now();
int** E = BucketSort(A,14);
auto stop2 = high_resolution_clock::now();
auto duration2 = duration_cast<nanoseconds>(stop2 - start2);
for (int i=0; i<10; i++){
for (int j = 0; E[i][j]>=0; j++){
lucky = E[i][j];
}
}
unlucky = -1;
for (int i=0; unlucky<0; i++){
for (int j = 0; E[i][j]>=0; j++){
unlucky = E[i][j];
break;
}
}
printlucky(AO, lucky, unlucky, n);
cout << setw(8) << left << n << setw(8) << left << duration.count();
cout << setw(8) << left << duration1.count() << setw(8) << left << duration2.count()<<endl;
}
// c1
int RandomSelect(int* A, int p, int r, int i){
if (p == r){
return A[p];
}
int q = RandomPartition(A,p,r);
input
CountingSort
lucky number is 4 ; unlucky number is 9 .
probabilities of a lucky winner is 12%; probabilities of a unlucky winner is 3%.
BucketSort
lucky number is 2 ; unlucky number is 9 .
probabilities of a lucky winner is 10%; probabilities of a unlucky winner is 3%.
100 860 880 2460
Type here to search
9:41 PM
4/20/2020


Online C++ Compiler - online ed
onlinegdb.com/online_c++_compiler
OnlineGDB beta
online compiler and debugger for c/c++
code. compile. run. debug. share.
IDE
My Projects
Learn Programming
Programming Questions
Jobs new
Sign Up
Login
27.5K
slack
Bring your team together with Slack, the collaboration hub for work.
ADS VIA CARBON
About • FAQ • Blog • Terms of Use • Contact Us • GDB Tutorial • Credits • Privacy
© 2016 - 2020 GDB Online
Run
Debug
Stop
Share
Save
{ } Beautify
Language C++
main.cpp
auto start2 = high_resolution_clock::now();
int** E = BucketSort(A,14);
auto stop2 = high_resolution_clock::now();
auto duration2 = duration_cast<nanoseconds>(stop2 - start2);
for (int i=0; i<10; i++){
for (int j = 0; E[i][j]>=0; j++){
lucky = E[i][j];
}
}
unlucky = -1;
for (int i=0; unlucky<0; i++){
for (int j = 0; E[i][j]>=0; j++){
unlucky = E[i][j];
break;
}
}
printlucky(AO, lucky, unlucky, n);
cout << setw(8) << left << n << setw(8) << left << duration.count();
cout << setw(8) << left << duration1.count() << setw(8) << left << duration2.count()<<endl;
}
// c1
int RandomSelect(int* A, int p, int r, int i){
if (p == r){
return A[p];
}
int q = RandomPartition(A,p,r);
input
probabilities of a lucky winner is 9.4%; probabilities of a unlucky winner is 6.6%.
BucketSort
lucky number is 8 ; unlucky number is 1 13 .
probabilities of a lucky winner is 9.2%; probabilities of a unlucky winner is 6.6%.
500 590 1310 2200
RandomSelect
Type here to search
9:41 PM
4/20/2020


Online C++ Compiler - online ed
onlinegdb.com/online_c++_compiler
OnlineGDB beta
online compiler and debugger for c/c++
code. compile. run. debug. share.
IDE
My Projects
Learn Programming
Programming Questions
Jobs new
Sign Up
Login
27.5K
slack
Bring your team together with Slack, the collaboration hub for work.
ADS VIA CARBON
About • FAQ • Blog • Terms of Use • Contact Us • GDB Tutorial • Credits • Privacy
© 2016 - 2020 GDB Online
Run
Debug
Stop
Share
Save
{ } Beautify
Language C++
main.cpp
auto start2 = high_resolution_clock::now();
int** E = BucketSort(A,14);
auto stop2 = high_resolution_clock::now();
auto duration2 = duration_cast<nanoseconds>(stop2 - start2);
for (int i=0; i<10; i++){
for (int j = 0; E[i][j]>=0; j++){
lucky = E[i][j];
}
}
unlucky = -1;
for (int i=0; unlucky<0; i++){
for (int j = 0; E[i][j]>=0; j++){
unlucky = E[i][j];
break;
}
}
printlucky(AO, lucky, unlucky, n);
cout << setw(8) << left << n << setw(8) << left << duration.count();
cout << setw(8) << left << duration1.count() << setw(8) << left << duration2.count()<<endl;
}
// c1
int RandomSelect(int* A, int p, int r, int i){
if (p == r){
return A[p];
}
int q = RandomPartition(A,p,r);
input
probabilities of a lucky winner is 8.4%; probabilities of a unlucky winner is 6%.
1000 870 1650 5730
...Program finished with exit code 0
Press ENTER to exit console.
Type here to search
9:41 PM
4/20/2020