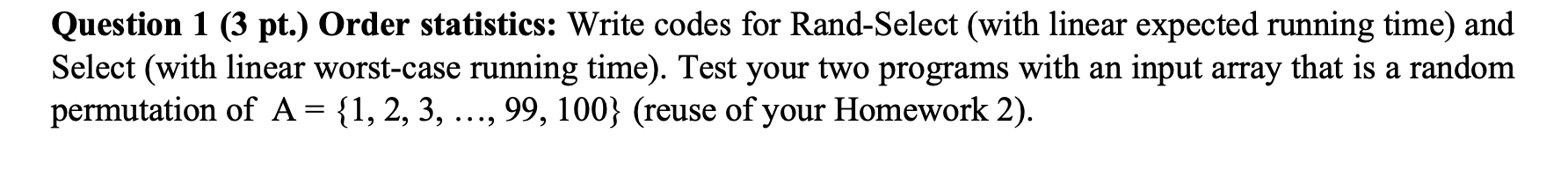
**HOMEWORK 3**

**JinjingMiao**



RandSelectLinerTime

#include<iostream>

#include<climits>

#include<cstdlib>

using namespace std;

int randomPartition(int arr[], int m, int r);

int kthSmallest(int arr[], int m, int r, int k)

{

if (k > 0 && k <= r - m + 1)

{

int pos = randomPartition(arr, m, r);

if (pos-m == k-1)

return arr[pos];

if (pos-m > k-1)

return kthSmallest(arr, m, pos-1, k);

return kthSmallest(arr, pos+1, r, k-pos+m-1);

}

return INT\_MAX;

}

void swap(int \*a, int \*b)

{

int temp = \*a;

\*a = \*b;

\*b = temp;

}

int partition(int arr[], int m, int r)

{

int x = arr[r], i = m;

for (int j = m; j <= r - 1; j++)

{

if (arr[j] <= x)

{

swap(&arr[i], &arr[j]);

i++;

}

}

swap(&arr[i], &arr[r]);

return i;

}

int randomPartition(int arr[], int m, int r)

{

int n = r-m+1;

int pivot = rand() % n;

swap(&arr[m + pivot], &arr[r]);

return partition(arr, m, r);

}

int main()

{

int i,j,r,temp;

int n=100;

srand((unsigned)time(NULL));

int arr[n];

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

arr[i] = i + 1;

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

{

r = rand() % n;

temp = arr[r];

arr[r] = arr[j];

arr[j] = temp;

}

cout << "Unsorted Aay is \n";

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

cout<<arr[i]<<" ";

cout<<"\n";

int k=99;

int kth = kthSmallest(arr,0,99,k);

cout << "the 99th smallest is " << kth << endl;

return 0;

}

OUTPUT :

Unsorted Aay is

42 40 68 7 67 62 4 95 36 19 94 11 78 17 26 48 13 39 66 30 84 27 88 89 31 51 28 52 57 6 46 32 76 71 41 50 23 75 64 15 9 45 16 2 59 35 20 44 1 54 65 96 37 83 5 79 70 53 98 34 60 21 43 38 72 85 80 93 18 74 47 63 22 29 49 86 87 3 10 82 100 97 58 25 12 99 55 81 33 90 56 92 14 91 69 77 73 8 24 61

the 99th smallest is 99

[Finished in 0.6s]

Select worst time

#include<iostream>

#include<climits>

#include<cstdlib>

using namespace std;

int randomPartition(int arr[], int m, int r);

int kthSmallest(int arr[], int m, int r, int k)

{

if (k > 0 && k <= r - m + 1)

{

int pos = randomPartition(arr, m, r);

if (pos-m == k-1)

return arr[pos];

if (pos-m > k-1)

return kthSmallest(arr, m, pos-1, k);

return kthSmallest(arr, pos+1, r, k-pos+m-1);

}

return INT\_MAX;

}

void swap(int \*a, int \*b)

{

int temp = \*a;

\*a = \*b;

\*b = temp;

}

int partition(int arr[], int m, int r)

{

int x = arr[r], i = m;

for (int j = m; j <= r - 1; j++)

{

if (arr[j] <= x)

{

swap(&arr[i], &arr[j]);

i++;

}

}

swap(&arr[i], &arr[r]);

return i;

}

int randomPartition(int arr[], int m, int r)

{

int n = r-m+1;

int pivot = rand() % n;

swap(&arr[m + pivot], &arr[r]);

return partition(arr, m, r);

}

int main()

{

int i,j,r,temp;

int n=100;

srand((unsigned)time(NULL));

int arr[n];

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

arr[i] = i + 1;

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

{

r = rand() % n;

temp = arr[r];

arr[r] = arr[j];

arr[j] = temp;

}

cout << "Unsorted Aay is \n";

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

cout<<arr[i]<<" ";

cout<<"\n";

int k=99;

int kth = kthSmallest(arr,0,99,k);

cout << "the 99th smallest is " << kth << endl;

return 0;

}

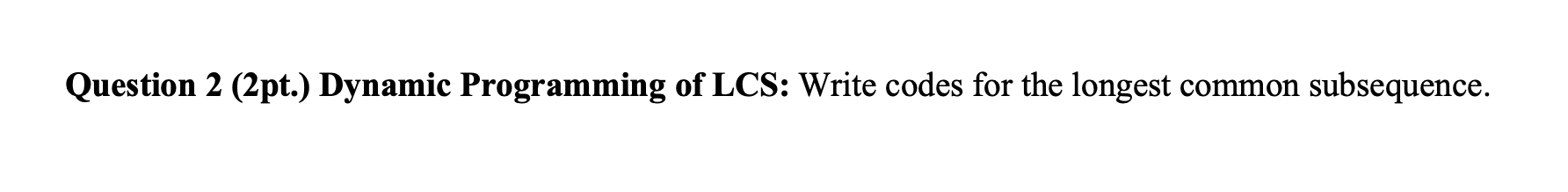
OUTPUT:

Unsorted Aay is

41 56 12 79 60 18 85 98 44 96 92 93 1 25 4 50 20 72 3 42 21 19 55 14 28 26 67 57 38 69 86 97 76 29 75 73 80 33 36 43 46 5 66 100 90 22 52 63 99 34 15 62 7 68 87 71 54 91 24 10 45 64 2 65 89 6 95 77 47 32 83 49 88 30 81 40 59 94 51 16 13 84 58 39 53 27 74 8 48 11 70 35 17 82 78 37 61 23 9 31

the 99th smallest is 99

[Finished in 0.4s]



#include <iostream>

#include <string>

#include <vector>

using namespace std;

int max(int a, int b)

{

return (a>b)? a:b;

}

// return lcs

int lcs(string &X, string &Y, int m, int n)

{

//（m+1)\*(n+1）

vector<vector<int> > table(m+1,vector<int>(n+1));

for(int i=0; i<m+1; ++i)

{

for(int j=0; j<n+1; ++j)

{

// column and row -> 0

if (i == 0 || j == 0)

table[i][j] = 0;

else if(X[i-1] == Y[j-1])

table[i][j] = table[i-1][j-1] + 1;

else

table[i][j] = max(table[i-1][j], table[i][j-1]);

}

}

return table[m][n];

}

int main()

{

string X = "ABCBDAB";

string Y = "BDCABA";

cout << "The length of LCS is " << lcs(X, Y, X.length(), Y.length());

cout << endl;

getchar();

return 0;

}

Output : The length of LCS is 4