Design and Analysis of Algorithms

Week-6

Name: Musali Reddy Manish Reddy

Roll.no :CH.SC.U4CSE24129

Quick sort:

(starting number as pivot element)

Code:

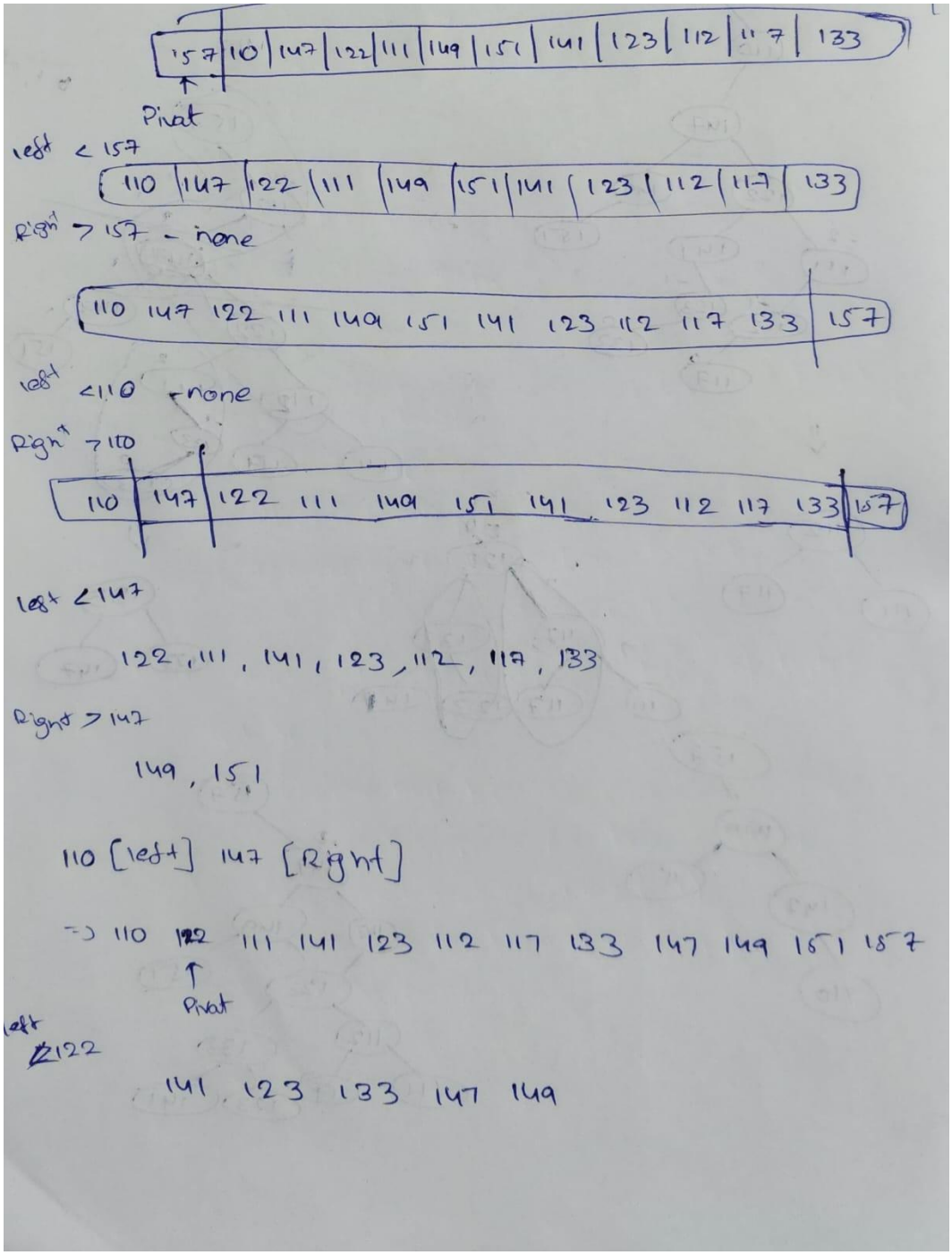
```
//CH.SC.U4CSE24129
#include <stdio.h>
int partitionFirst(int a[], int low, int high)
{
   int pivot, i, j, temp;
   pivot = a[low];
   i = low + 1;
   j = high;

   while (i <= j)
   {
      while (a[i] <= pivot && i <= high)
         i++;
      while (a[j] > pivot)
         j--;

      if (i < j)
      {
         temp = a[i];
         a[i] = a[j];
```

```
            a[j] = temp;
        }
    }
    temp = a[low];
    a[low] = a[j];
    a[j] = temp;

    return j;
}
void quickSortFirst(int a[], int low, int high)
{
    int p;
    if (low < high)
    {
        p = partitionFirst(a, low, high);
        quickSortFirst(a, low, p - 1);
        quickSortFirst(a, p + 1, high);
    }
}
int main()
{
    int a[12] = {157,110,147,122,111,149,151,141,123,112,117,133};
    int i;
    quickSortFirst(a, 0, 11);
    printf("Sorted Array:\n");
    for (i = 0; i < 12; i++)
        printf("%d ", a[i]);
    return 0;
}
```

Output:

```
Sorted Array:
110 111 112 117 122 123 133 141 147 149 151 157
--------------------------------
```

Handwritten:



110 [left] 147 [right] 157

left < 122

111 112 117

Right > 122

~~123~~ 141 123 133 149

110 [left] 122 [Right]

110 111 112 117 122 | 141 123 133 147 149 151 157

Pivot

< 141

123 133

> 141

147 149 151 157

110 111 112 117 122 123 133 141 | 147 149 151 157

Pivot

< 147 - none

> 147

149 151 157

110 111 112 117 122 123 133 141 147 | 149 151 157

Pivot

left Pivot Righ

< Piv

Pivot

```
#include <stdio.h>
int PartitionFirst (int a[], int low, int high) {
    int Pivot, i, j, temp;
    Pivot = a[low];
    i = low+1;
    j = high;
    while (i<=j) {
      while (a[i] <= Pivot && i<= high)
        i++;
      while (a[j] > Pivot)
        j--;
      if (i<j) {
        temp = a[i];
        a[i] = a[j];
        a[j] = temp;
      }
    }
    temp = a[low];
    a[low] = a[j];
    a[j] = temp;
    return j;
}
void quickSortFirst(int a[], int low, int high) {
    int P;
    if (low < high) {
      P = PartitionFirst(a, low, high);
      quickSortFirst (a, low, P-1);
      quickSortFirst (a, P+1, high);
    }
}
```

```
int main () {
    int a[12] = {157, 110, 147, 122, 111, 149, 151, 141, 123, 112,
                 117, 133};
    int i;
    quickSortFirst (a, 0, 11);
    Printf("Sorted Array: \n");
    for (i=0; i<12; i++)
        Printf ("%d ", a[i]);
    return 0;
}
```

Quick sort:

(last number as pivot element)

Code:

```
//CH.SC.U4CSE24129
#include <stdio.h>
int partitionLast(int a[], int low, int high)
{
   int pivot, i, j, temp;
   pivot = a[high];
   i = low - 1;
   for (j = low; j < high; j++)
   {
      if (a[j] <= pivot)
      {
         i++;
         temp = a[i];
         a[i] = a[j];
         a[j] = temp;
      }
   }
   temp = a[i + 1];
   a[i + 1] = a[high];
   a[high] = temp;
   return i + 1;
}
void quickSortLast(int a[], int low, int high)
{
   int p;
```

```
    if (low < high)
    {
        p = partitionLast(a, low, high);
        quickSortLast(a, low, p - 1);
        quickSortLast(a, p + 1, high);
    }
}
int main()
{
    int a[12] = {157,110,147,122,111,149,151,141,123,112,117,133};
    int i;
    quickSortLast(a, 0, 11);
    printf("Sorted Array:\n");
    for (i = 0; i < 12; i++)
        printf("%d ", a[i]);
    return 0;
}
```

Output:

```
Sorted Array:
110 111 112 117 122 123 133 141 147 149 151 157
--------------------------------
```

Handwritten:

Last Element as PIVOT

157 110 147 122 111 149 151 141 123 112 117 133

↑ Pivat

122 < 133

110 157 147 122 111 149 151 141 123 112 117 133

110 122 147 157 111 149 151 141 123 112 117 133

111 < 133

110 122 111 157 147 149 151 141 123 112 117 133

123 < 133

110 122 111 123 147 149 151 141 157 112 117 133

112 < 133

117 < 133

110 122 111 123 112 117 151 141 157 147 149 133

```
#include <stdio.h>
int PartitionLast (int a[], int low, int high) {
    int Pivot, i, j, temp;
    Pivot = a[high];
    i = low - 1;
    for (j = low; j < high; j++) {
        if (a[j] <= Pivot) {
            i++;
            temp = a[i]
            a[i] = a[j]
            a[j] = temp;
        }
    }
    temp = a[i+1];
    a[i+1] = a[high];
    a[high] = temp;
    return i+1;
}
void quicksortlast (int a[], int low, int high) {
    int P;
    if (low < high) {
        P= Partitionlast (a, low, high);
        quicksortlast (a, low, P-1);
        quicksortlast (a, P+1, high);
    }
}
intmain() {
    int a[12] = {157, 110, 147, 122, 111, 149, 151, 141, 123, 112, 117,
    int i;
    quicksortlast (a, 0, 11);
    Printf ("sorted Array :\n");
    for (i=0; i<12; i++)
        Printf ("%d", a[i]);
    return 0;
}
```

Quick sort:

(Middle number as pivot element)

Code:

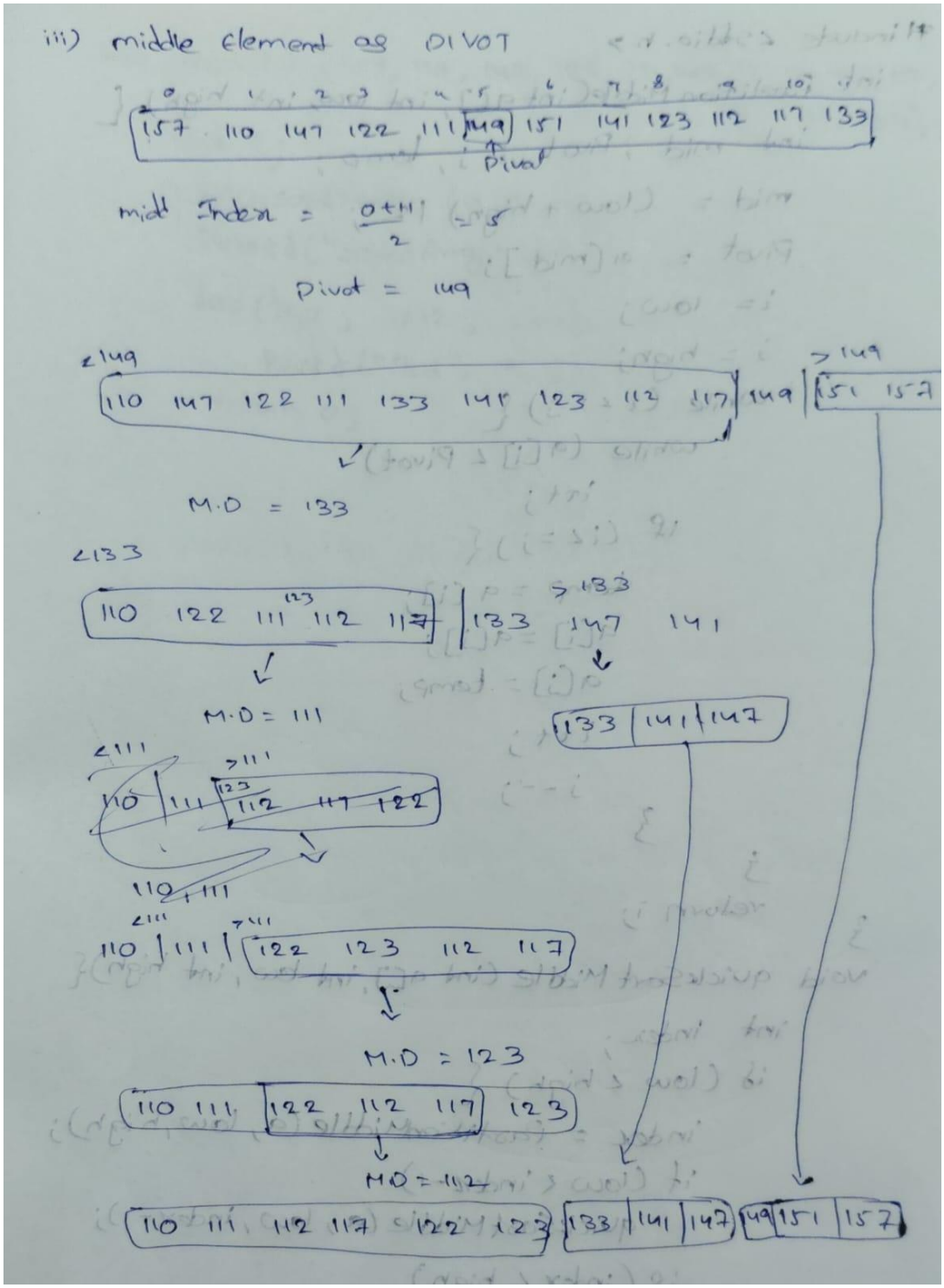
```
//CH.SC.U4CSE24129
#include <stdio.h>
int partitionMiddle(int a[], int low, int high)
{
   int mid, pivot, i, j, temp;
   mid = (low + high) / 2;
   pivot = a[mid];
   i = low;
   j = high;
   while (i <= j)
   {
      while (a[i] < pivot)
         i++;

      while (a[j] > pivot)
         j--;

      if (i <= j)
      {
         temp = a[i];
         a[i] = a[j];
         a[j] = temp;

         i++;
         j--;
      }
```

```
    }
    return i;
}
void quickSortMiddle(int a[], int low, int high)
{
    int index;
    if (low < high)
    {
        index = partitionMiddle(a, low, high);
        if (low < index - 1)
            quickSortMiddle(a, low, index - 1);
        if (index < high)
            quickSortMiddle(a, index, high);
    }
}
int main()
{
    int a[12] = {157,110,147,122,111,149,151,141,123,112,117,133};
    int i;
    quickSortMiddle(a, 0, 11);
    printf("Sorted Array:\n");
    for (i = 0; i < 12; i++)
        printf("%d ", a[i]);
    return 0;
}
```

Output:

```
Sorted Array:
110 111 112 117 122 123 133 141 147 149 151 157
--------------------------------
```

Handwritten:



```
#include <stdio.h>
int PartitionMiddle(int a[], int low, int high) {
    int mid, Pivot, i, j, temp;
    mid = (low + high) / 2;
    Pivot = a[mid];
    i = low;
    j = high;
    while (i <= j) {
        while (a[i] < Pivot)
            i++;
        if (i <= j) {
            temp = a[i];
            a[i] = a[j];
            a[j] = temp;
            i++;
            j--;
        }
    }
    return i;
}
void quickSortMiddle(int a[], int low, int high){
    int index;
    if (low < high) {
        index = PartitionMiddle(a, low, high);
        if (low < index-1)
            quickSortMiddle(a, low, index-1);
        if (index < high)
            quickSortMiddle(a, index, high);
    }
}
```

```
int a[12] = {157, 110, 147, 122, 111, 149, 151, 141, 123, 112,
             117, 133};
    int i;
    quickSortMiddle(a, 0, 11);
    Printf("Sorted Array: \n");
    for (i=0; i<12; i++)
        Printf("%d", a[i]);
    return 0;
}
```