ADA: QUICK SORT

0	1	2	3	4	5	6
→ 26	19	12	22	33	35	29

n = 7

```
Quicksort(0, 6)
          L  h
{
    if(0<6) - T
    {
        partition(0, 6)
                  L  h
        {
            pivot = a[0] = 26 , i = 0 , j = 6
            while(0<=6) - T
            {
                while(26<=26 && 0<=6) - T
                        i = 1;
                while(19<=26 && 1<=6) - T
                        i = 2;
                while(12<=26 && 2<=6) - T
                        i = 3;
                while(22<=26 && 3<=6) - T
                        i = 4
                while(33<=26 && 4<=6) - F
                while(29>26 && 6>=0) - T
                        j = 5;
                while(35>26 && 5>=0) - T
                        j = 4;
                while(33>26 && 4>=0) - T
                        j = 3
                while(22>26) - F
                if(4<3) - F
                else { temp = a[0] = 26
                       a[0] = a[3] = 22
                       a[3] = temp = 26
                     }
            }
            while(4<=3) - F
        }
        return 3;
    }
```

SHIVAS

0	1	2	3	4	5	6
#22	19	12	26	33	35	29

Quicksort (0, 2)
{
if(0<2) - T
{
partition (0, 2)
{
Pivote = a[0] = 22, i=0, j=3
while(0<=2)=T
{
while(22<=22 && 0<=2) - T
i=1;
while[19<=22 && 1<=2) - T
i=2;
while(12<=22 && 2<=2)- T
i=3;
while(26<=22) - F
while(33>=22) - F
if(3<2)-F
else { temp = a[0] = 22
a[0] = a[2] = 12
a[2] = temp = 22
}
while(3<=2) - F
return 2;
}

0	1	2	3	4	5	6
#12	19	22	26	33	35	29

Quicksort(0, 1)
{
if(0<1) -T
{
partition(0, 1)
{
pivote = a[0] = 12, i=0 j=1
while(0<=1)-T
{
while(12<=12 && 0<=1)-T
i=1;
while(19<=12) - F
while(19>12 && 1>=0)-T
j=0;
while(12>12) - F
if(1<0) - F
else { temp = a[0] = 12
a[0] = a[0] = 12
a[0] = temp = 12
}
while(1<=0)-F

}. return 0;

#	12	19	22	26	33	35	29
	0	1	2	3	4	5	6

Quicksort (1, -1)
{ L h
if (1 <= -1) - F
}
Quicksort (1, 1)
{ L h
if (1 <= 1) - T
}
}
Quicksort (3, 2)
{ L h
if (3 <= 2) - F
}
}
Quicksort (4, 6)
{ L h
if (4 <= 6) - T
{
partition (4, 6)
{ L h
pivot = a[4] = 33, i = 4, j = 6
while (4 <= 6) - T
{
while (33 <= 33 && 4 <= 6) - T
i = 5;
while (35 <= 33) - F
while (29 > 33) - F
if (5 <= 6) - T
{
temp = a[5] = 35
a[5] = a[6] = 29
a[6] = temp = 35

#	12	19	22	26	33	29	35
	0	1	2	3	4	5	6

while (5 <= 6) - T
{ i j
while (29 <= 33) - F

while (29 <= 33 && 5 <= 6) - T

i = 6;

while (35 <= 33) - F

while (35 > 33 && 6 >= 4) - T

j = 5;

while (29 > 33) - F

if (6 < 5) - F

else { temp = a[4] = 33

a[4] = a[5] = 29

a[5] = 33

}

}

while (6 < 5) - F

return 5;

}

Quicksort (4, 4)

{# 12 19 22 26 29 33 35

if (4 < 4) - F

}

Quicksort (6, 6)

{

if (6 < 6) - F

}

}

Result of Quicksort

12 19 22 26 29 35

Quicksort Tree :