

LAB-3

LinearBinaryTime.c

```
#include <stdio.h>

#include <time.h>

#include <stdlib.h>

#include <math.h>

int linears(int i);

int binaryS(int,int);

void selectionsort(int *a,int n);

int a[1000000],key,ch;


int main()

{

    int i,b,n;

    clock_t start,end;

    do

    {

        printf("\nchoose method\n");

        printf("1.recursive linear search\n2.linear binary search\n3.exit\n");

        scanf("%d",&ch);

        switch(ch)
```

```
{  
    case 1:  
        printf("enter number of elements-\n");  
        scanf("%d",&n);  
        printf("%d numbers generated-\n",n);  
        for(i=0;i<=n;i++)  
        {  
            a[i]=rand()%10000;  
            printf("%d\t", a[i]);  
        }  
        printf("\nEnter the number to be searched\n");  
        scanf("%d",&key);  
  
        start=clock();  
        b=linears(0);  
        end=clock();  
        printf("Time taken:%f\n",(((double)(end-start))/CLOCKS_PER_SEC));  
        if(b==5)  
            printf("Number not found\n");  
        else  
            printf("Number found at position: %d\n",b+1);  
        break;
```

case 2:

```
printf("enter number of elements-\n");
```

```
scanf("%d",&n);
```

```
printf("%d numbers generated-\n",n);
```

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

```
{
```

```
    a[i]=rand()%10000;
```

```
    printf("%d\t", a[i]);
```

```
}
```

```
printf("\nEnter the number to be searched\n");
```

```
scanf("%d",&key);
```

```
start=clock();
```

```
selectionsort(a,n);
```

```
printf("The elements after sorting are :");
```

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

```
        printf("%d ",a[i]);
```

```
b=binaryS(0,n-1);
```

```
end=clock();
```

```
printf("\nTime taken:%f\n",(((double)(end-start))/CLOCKS_PER_SEC));
```

```
    if(b== -1)
    {
        printf("Number not found\n");
    }
    else
    {
        printf("Number %d found at position:%d\n",key,(b+1));
    }

    break;
default: break;
}
}while(ch!=3);
return 0;
}
```

```
int linears(int i)
{
    if(i==5)
        return 5;
    else if(a[i]==key)
        return i;
    else
```

```
    linears(++i);  
}
```

```
int binaryS(int f,int l)  
{  
    int m;  
    m=((f+l)/2);  
    if(key==a[m])  
    {  
        return m;  
    }  
    else if(key>a[m])  
    {  
        return binaryS(++m,l);  
    }  
    else if(key<a[m])  
    {  
        return binaryS(f,--m);  
    }  
    else if(f>l)  
    {
```

```
        return -1;
    }
}
```

```
void selectionsort(int *a,int n)
```

```
{
    int temp,pos,small;
    for(int i=0;i<=n-2;i++)
    {
        small=a[i];
        pos=i;
        for(int j=i+1;j<=n-1;j++)
        {
            if(a[j]<small)
            {
                small=a[j];
                pos=j;
            }
        }
        temp=a[i];
        a[i]=a[pos];
        a[pos]=temp;
    }
}
```

}

}