

Name: Manasi. B. Kshirsagar

PRN: ST7111009

Roll No.: 06

Class: BE Comp SS

Code (Binary Search):

```
#include<iostream>
```

```
#include<algorithm>
```

```
#include<stdlib.h>
```

```
#include<omp.h>
```

```
#include<ctime>
```

```
using namespace std;
```

```
int binary(int *, int, int, int);
```

```
int binary(int *a, int low, int high, int key)
```

```
{
```

```
    int mid;
```

```
    mid=(low+high)/2;
```

```
    int low1,low2,high1,high2,mid1,mid2,found=0,loc=1;
```

```
    #pragma omp parallel sections
```

```
    {
```

```
        #pragma omp section
```

```
        {
```

```
            low1=low;
```

```
            high1=mid;
```

```

while(low<=high)
{
    if(!(key>=a[low] && key<=a[high]))
    {
        low=low+high;
        continue;
    }
    mid=(low+high)/2;
    if(key==a[mid])
    {
        found=1;
        loc=mid;
        low=high+1;
    }

    else if(key>a[mid])
    {
        low=mid+1;
    }
    else if(key<a[mid])
        high=mid-1;
    }
}

```

```

#pragma omp section

```

```

{

```

```

low2=mid+1;
high2=high;
while(low2<=high2)
{
    if(!(key>=a[low2] && key<=a[high2]))
    {
        low2=low2+high2;
        continue;
    }
    mid2=(low2+high2)/2;

    if(key==a[mid2])
    {
        found=1;
        loc=mid2;
        low2=high2+1;
    }
    else if(key>a[mid2])
    {

        low2=mid2+1;
    }
    else if(key<a[mid2])
        high2=mid2-1;

}

```

```

        }
    }
    return loc;
}

```

```

int main()
{
    int *a,i,n,key,loc=1;
    cout<<"\n Enter Total no of Elements=>";
    cin>>n;
    a=new int[n];

    for(i=0;i<n;i++)
    {
        a[i]=rand()%(2*n);
    }

    sort(a,a+n);
    cout<<"Elements are : "<<endl;
    for(i=0;i<n;i++)
    {
        cout<<a[i]<<" ";
    }

    cout<<endl;
    clock_t start=clock();
    cout<<"\n Enter Key to find=>";
}

```

```

cin>>key;

loc=binary(a,0,n-1,key);

if(loc==1)

    cout<<"\n Key not found.";

else

    cout<<"\n Key found at position=>"<<loc+1;

clock_t end=clock();

double time = (end-start)/CLOCKS_PER_SEC;

cout<<"\n Time: "<<time<<"ms"<<endl;

return 0;

}

```

Output:

The screenshot shows the Visual Studio Code interface with a C++ project named 'binsearch.cpp - HPC - Visual Studio Code'. The Explorer pane on the left shows the project structure with files like 'binsearch.cpp', 'soln.cpp', 'addn.cpp', 'BFS.cpp', 'binsearch.cpp', 'main.cpp', 'soln.cpp', and 'vectmultiply.cpp'. The main editor displays the 'binsearch.cpp' file, which contains a binary search function and a main function. The terminal at the bottom shows the output of the program, which is a large list of numbers from 0 to 19, followed by the prompt 'Enter Key to find->942'. The output also shows 'Key found at position->587' and 'Time: 2ms'.

```

0 333 334 335 343 350 353 355 355 355 356 359 368 362 363 365 368 369 370 372 380 382 383 386 389 391 392 393 404 409 410 411 413 413 413 414 416 418 421 422 423 423 423 424
426 428 428 429 433 439 439 439 443 450 455 460 463 464 464 466 466 467 467 467 472 472 474 474 476 477 477 480 481 482 483 483 484 484 485 485 488 488 489 491 492 493 500
503 503 504 508 510 511 512 519 519 520 523 525 526 527 529 532 534 536 537 538 540 540 541 541 548 540 549 549 550 555 555 556 556 559 565 565 570 576 580 584 585 588 589 591 593 5
96 596 598 600 600 601 604 604 606 607 608 608 609 610 617 617 618 619 623 625 625 625 626 626 634 634 636 637 637 639 641 646 648 648 649 650 651 651 652 654 658 662 662 663 66
4 667 668 670 671 673 674 675 678 678 681 683 685 685 686 687 688 689 690 692 693 695 695 696 698 700 701 702 703 704 704 705 710 712 712 716 717 722 723 725 726 729 734
734 736 740 741 745 745 748 750 756 757 757 758 759 760 762 766 767 771 771 774 777 778 778 783 786 787 788 798 798 798 800 802 808 813 814 818 823 825 827 827 833 833 835 836
838 843 848 850 851 855 858 859 865 867 868 869 869 875 875 877 886 887 888 888 893 893 896 896 900 900 908 908 902 902 909 909 911 912 913 913 923 924 924 929 932 932 935 937 9
38 940 941 941 942 943 944 944 945 945 946 948 949 955 958 962 962 962 962 963 966 969 970 971 972 974 974 975 989 993 995 1001 1002 1003 1003 1006 1007 1008 1008 1010 1010 1011 101
5 1017 1018 1020 1021 1022 1023 1025 1030 1030 1031 1035 1035 1037 1038 1039 1040 1049 1060 1061 1064 1067 1071 1071 1072 1072 1073 1075 1075 1080 1086 1087 1088 1090 1093 1097
1098 1101 1102 1103 1107 1108 1109 1109 1110 1111 1113 1115 1117 1119 1124 1129 1131 1132 1141 1141 1142 1145 1150 1152 1152 1153 1156 1157 1161 1164 1168 1169 1169 1170 1173 11
74 1184 1185 1185 1186 1187 1189 1192 1192 1195 1195 1196 1196 1199 1200 1200 1200 1200 1205 1205 1213 1216 1221 1221 1221 1222 1224 1232 1240 1245 1247 1249 1253 1255 1258 1261
1262 1262 1264 1264 1264 1264 1269 1271 1281 1281 1282 1285 1286 1288 1290 1292 1297 1309 1313 1313 1314 1314 1316 1318 1318 1322 1323 1320 1333 1334 1337 1337 1342 1342 1
347 1348 1350 1350 1350 1353 1355 1357 1357 1357 1358 1359 1361 1361 1363 1369 1371 1374 1375 1376 1384 1391 1392 1398 1401 1402 1405 1410 1411 1416 1416 1421 1422 1423 1425 142
6 1430 1430 1432 1434 1436 1437 1441 1446 1447 1448 1451 1452 1454 1457 1457 1458 1459 1461 1462 1477 1478 1484 1487 1487 1489 1496 1497 1498 1503 1505 1506 1508 1510 1511 1512
1514 1515 1515 1518 1520 1527 1529 1535 1537 1538 1538 1539 1541 1542 1543 1543 1545 1546 1547 1547 1548 1548 1549 1556 1556 1557 1558 1561 1561 1565 1565 1571 1574 1576 15
77 1578 1584 1584 1585 1587 1588 1589 1589 1591 1591 1599 1600 1601 1601 1602 1605 1611 1616 1617 1619 1620 1622 1624 1624 1625 1626 1626 1627 1627 1629 1629 1629 1634
1635 1637 1643 1644 1646 1648 1650 1653 1655 1655 1657 1658 1659 1662 1667 1668 1673 1673 1676 1676 1679 1690 1694 1694 1699 1701 1705 1705 1706 1711 1711 1713 1718 1718 1721 1
723 1724 1724 1726 1728 1734 1734 1737 1741 1748 1752 1753 1753 1754 1756 1757 1758 1758 1759 1760 1763 1763 1763 1769 1773 1775 1781 1783 1786 1788 1789 1790 1790 1796 1801 180
5 1807 1811 1812 1813 1815 1815 1815 1818 1824 1824 1825 1829 1829 1829 1831 1831 1832 1832 1833 1840 1841 1842 1844 1844 1847 1850 1851 1853 1855 1855 1861 1864 1866 1869 1869
1869 1870 1870 1874 1878 1881 1881 1882 1885 1890 1892 1894 1895 1898 1900 1901 1902 1903 1905 1913 1913 1923 1924 1926 1928 1930 1931 1934 1936 1938 1938 1942 1944 1945 1948 19
49 1951 1954 1954 1956 1958 1958 1959 1961 1961 1964 1966 1971 1971 1972 1974 1976 1977 1977 1982 1985 1985 1986 1990 1992 1993 1994 1996 1997 1998 1999

```

Enter Key to find->942

Key found at position->587

Time: 2ms

PS E:\2k20\CPP\HPC>

Code (DFS):

```
#include<iostream>
```

```
#include<omp.h>
```

```
#include<chrono>
```

```
#include<time.h>
```

```
using namespace std;
```

```
using namespace std::chrono;
```

```
class Node
```

```
{
```

```
    public:
```

```
        int val;
```

```
        Node *left;
```

```
        Node *right;
```

```
        Node(int val)
```

```
        {
```

```
            this->val = val;
```

```
            this->left = NULL;
```

```
            this->right = NULL;
```

```
        }
```

```
};
```

```
void parallel_dfs(Node *t)
```

```

{
    if(t == NULL)
        return;
    #pragma omp parallel sections
    {
        #pragma omp section
        {
            parallel_dfs(t->left);
std::cout<<t->val<<" ";
        }
        #pragma omp section
        {
            parallel_dfs(t->right);
        }
    }
}

```

```

int main()
{
    int n = 10000;
    int arr[n];

    for(int i=0;i<n;i++)
        arr[i] = rand()%100;

    Node *root = NULL;

```

```

if(root == NULL)
{
    Node *node = new Node(arr[0]);
    root = node;
}

for(int i=1;i<n;i++)
{
    Node *n = new Node(arr[i]);
    Node *temp = root;

    while(temp != NULL)
    {using namespace std::chrono;
        if(temp->val > arr[i])
        {
            if(temp->left != NULL)
                temp = temp->left;
            else
            {
                temp->left = n;
                break;
            }
        }

        else

```



```

        {
            if(temp->right != NULL)
                temp = temp->right;
            else
            {
                temp->right = n;
                break;
            }
        }
    }
}

```

```

time_point<system_clock> start,end;

```

```

start = system_clock::now();

```

```

parallel_dfs(root);

```

```

end= system_clock::now();

```

```

cout<<endl;

```

```

duration<double> t = end-start;

```

```

    cout<<"The time required for parallel is :- "<<t.count()<<"ms"<<endl;

```

```

return 0;

```

```

}

```

Output:

