DSA

PROJECT

Bin packing

Presented by

Debprotim chakrabarti 17bce0408

Aniket gupta 17bce2018

Aayush sahay 17bce2141

ABSTRACT

We have taken two real time problems in this project. The first is overseas cargo shipment which has been implemented using best-fit algorithm of bin packing. The second is passenger ferry boat which has been implemented using next-fit algorithm of bin packing.

INTRODUCTION

Two real time problems are-

1.Passenger ferry boat – Passenger ferry boats are used very much by tourists and is a very popular business. Therefore, to make their working efficient, we have decided to write a code using bin packing. In this, we accommodate a number of families or groups of people travelling together using next-fit algorithm of bin packing. If a certain group or family is to be accommodated on the ship, first the program checks whether there is space for them or not. If there is space for them that ship is automatically assigned to them but if there is not enough space for the whole family then according to the program the family is immediately assigned another boat. This way the family stays together to enjoy and no delay occurs. Therefore, it is beneficial for both the ship companies as well as the tourists.

2.Overseas Cargo shipment – As we know, overseas shipment costs very much. To increase profits, companies have to find the most efficient way to ship their materials so that more materials are shipped in one ship. To solve this real-life problem, we have decided to write a code using best-fit algorithm of bin packing. It is to be noted that the materials to be loaded on the ship come in random manner, therefore once loaded it cannot be unloaded as that would be very hectic and difficult. Firstly, materials are loaded on the ship. Now suppose another set of materials has to be loaded

which would exceed the total weight capacity of the ship as other materials are also present, the program will automatically assign another ship to the new lot of materials. Now suppose another set of materials has to be loaded which can be accommodated in both the ships, so according to our program this new lot of materials will be assigned to the ship which will have least space left out of the other ships after the loading of the new lot of materials.

BASE PAPER EXPLANATION –

Let j denote the current item, and s the last created shelf:

• Next-Fit Decreasing Height (NFDH) strategy: item j is packed left justified on shelf s, if it fits. Otherwise, a new shelf (s := s+1) is created, and j is packed left justified into it.

• Best-Fit Decreasing Height (BFDH) strategy: item j is packed left justified on that shelf, among those where it fits, for which the unused horizontal space is a minimum. If no shelf can accommodate j, a new shelf is initialized as in NFDH.

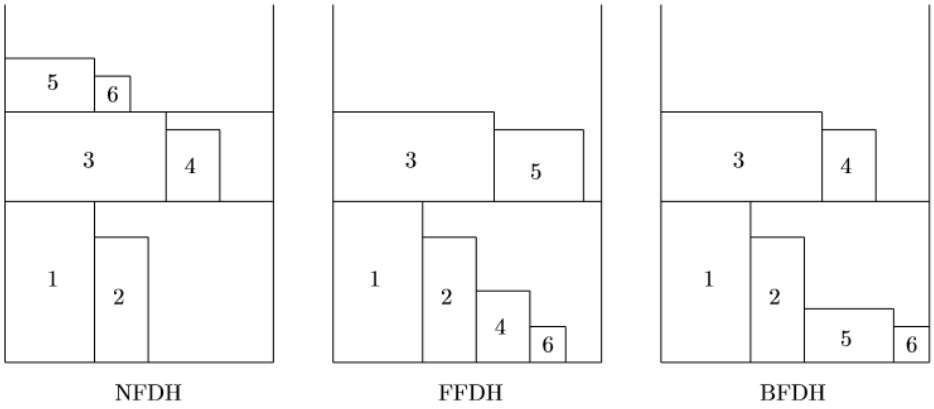


Figure 2.1: Shelf packing strategies.

NFDH – Next-Fit Algorithm

BFDH – Best-Fit Algorithm

IMPLEMENTATION CODE AND SCREEN SHOTS

FOR PASSENGER FERRY BOAT WE USED NEXT FIT BIN PACKING :

THE CODE IS AS FOLLOWS

```
//topic is PASSENGER FERRY BOAT
// C++ program to find number of ships required using
// next fit algorithm.
#include<iostream>
#include<time.h>
using namespace std;

// Returns number of ships required using next fit
int nextFit(int capacity[], int n, int c)
{

   int temp,temp2;
   int res=1;
   temp=c;
```

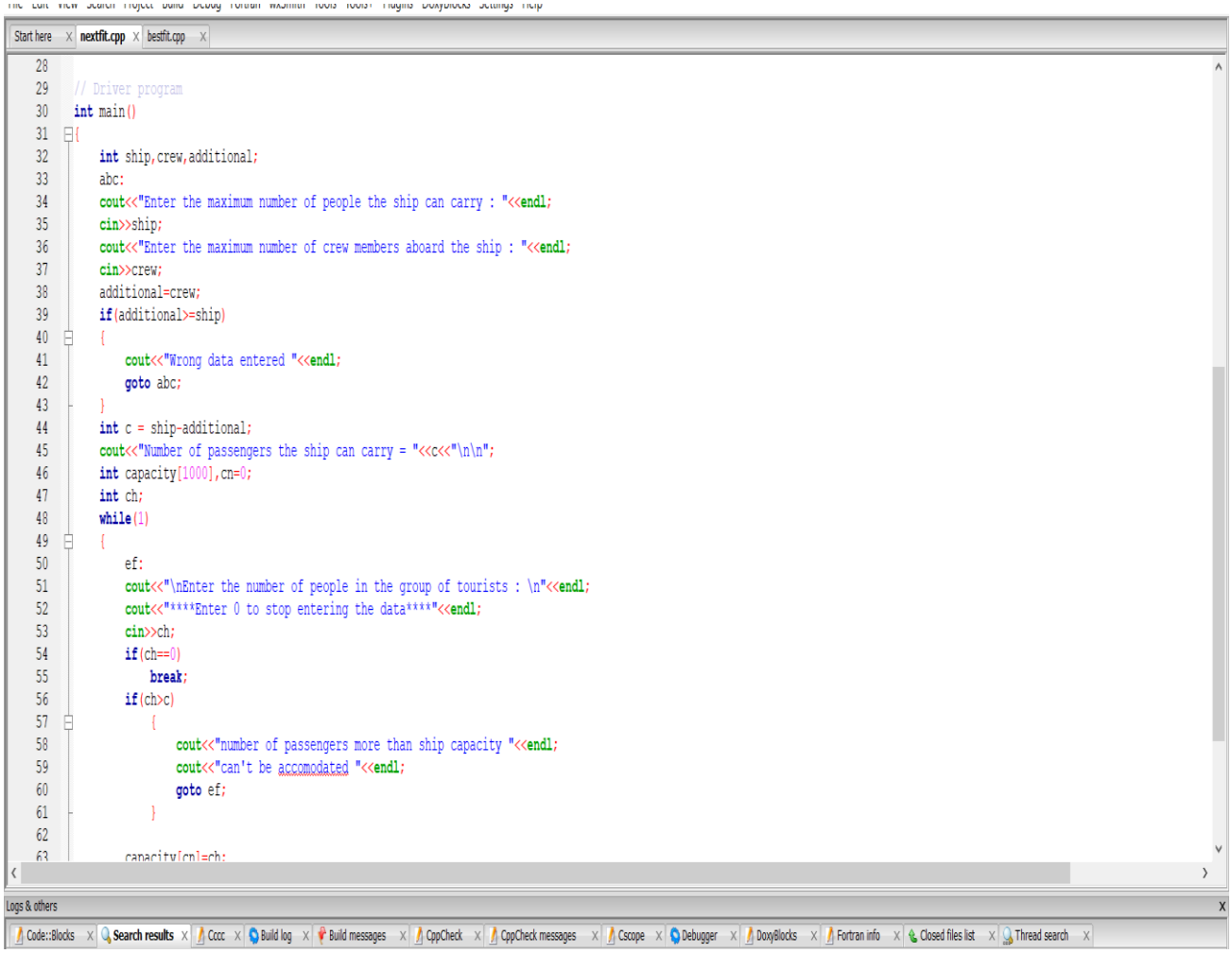
```
    for(int i=0;i<n;i++)
    {
        c=c-capacity[i];
        if(c<0)
        {
            res++;
            c=temp;
            i=i-1;
        }

    }
    return res;
}

// Driver program
int main()
{
    int ship,crew,additional;
    abc:
    cout<<"Enter the maximum number of people
the ship can carry : "<<endl;
    cin>>ship;
```

```
    cout<<"Enter the maximum number of crew
members aboard the ship : "<<endl;
    cin>>crew;
    additional=crew;
    if(additional>=ship)
    {
        cout<<"Wrong data entered "<<endl;
        goto abc;
    }
    int c = ship-additional;
    cout<<"Number of passengers the ship can carry
= "<<c<<"\n\n";
    int capacity[1000],cn=0;
    int ch;
    while(1)
    {
        ef:
        cout<<"\nEnter the number of people in the
group of tourists : \n"<<endl;
        cout<<"****Enter 0 to stop entering the
data****"<<endl;
        cin>>ch;
```

```
        if(ch==0)
            break;
        if(ch>c)
            {
                cout<<"number of passengers more than
ship capacity "<<endl;
                cout<<"can't be accomodated "<<endl;
                goto ef;
            }

        capacity[cn]=ch;
        cn++;
    }
    int n = cn;
    cout << "Number of ships required in : "
         << nextFit(capacity, n, c);
    return 0;
}
```

Start here | nextfit.cpp | bestfit.cpp

```cpp
//topic is PASSENGER FERRY BOAT
// C++ program to find number of ships required using
// next fit algorithm.
#include<iostream>
#include<time.h>
using namespace std;

// Returns number of ships required using next fit
int nextFit(int capacity[], int n, int c)
{

   int temp,temp2;
   int res=1;
   temp=c;
   for(int i=0;i<n;i++)
   {
       c=c-capacity[i];
       if(c<0)
       {
           res++;
           c=temp;
           i=i-1;
       }

   }
   return res;
}

// Driver program
int main()
{
    int ship,crew,additional;
    abc:
    cout<<"Enter the maximum number of people the ship can carry : "<<endl;
    cin>>ship;
    cout<<"Enter the maximum number of crew members aboard the ship : "<<endl;
```


Start here
nextfit.cpp
bestfit.cpp
// Driver program
int main()
{
int ship,crew,additional;
abc:
cout<<"Enter the maximum number of people the ship can carry : "<<endl;
cin>>ship;
cout<<"Enter the maximum number of crew members aboard the ship : "<<endl;
cin>>crew;
additional=crew;
if(additional>=ship)
{
cout<<"Wrong data entered "<<endl;
goto abc;
}
int c = ship-additional;
cout<<"Number of passengers the ship can carry = "<<c<<"\n\n";
int capacity[1000],cn=0;
int ch;
while(1)
{
ef:
cout<<"\nEnter the number of people in the group of tourists : \n"<<endl;
cout<<"****Enter 0 to stop entering the data****"<<endl;
cin>>ch;
if(ch==0)
break;
if(ch>c)
{
cout<<"number of passengers more than ship capacity "<<endl;
cout<<"can't be accomodated "<<endl;
goto ef;
}
capacity[cn]=ch;
Logs & others
Code::Blocks
Search results
Cccc
Build log
Build messages
CppCheck
CppCheck messages
Cscope
Debugger
DoxyBlocks
Fortran info
Closed files list
Thread search

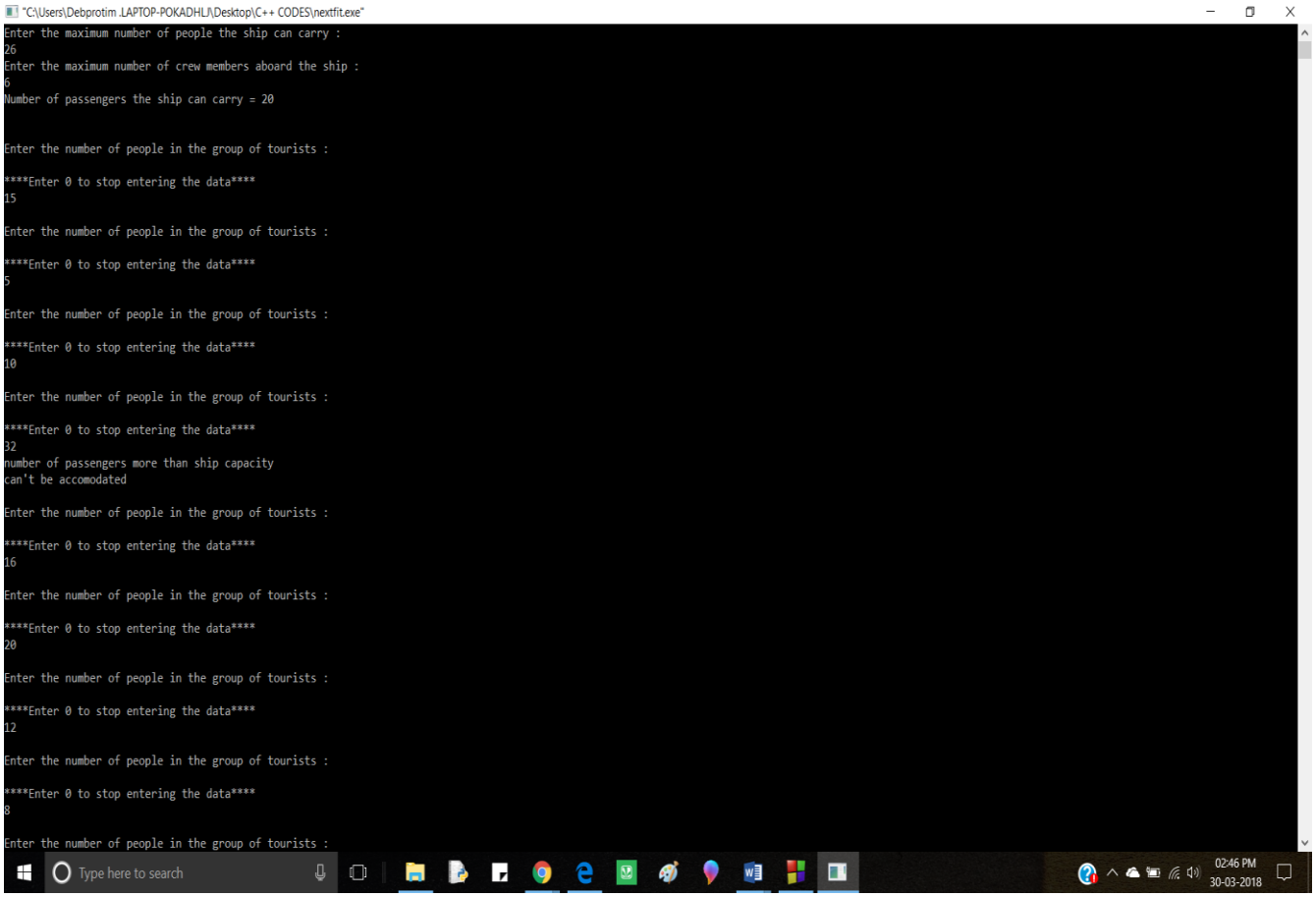
Start here | nextfit.cpp | bestfit.cpp

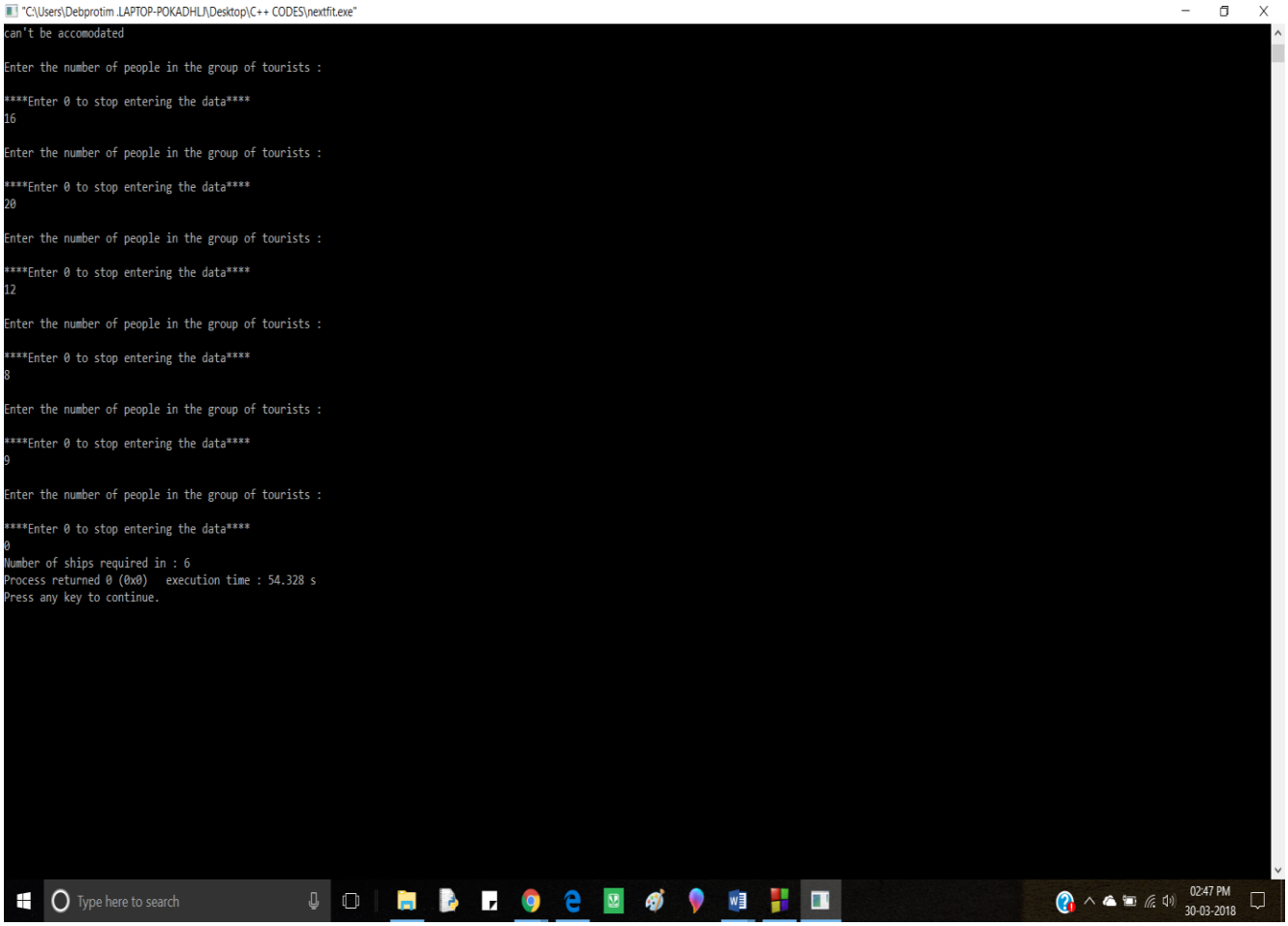
```
37      cin>>crew;
38      additional=crew;
39      if(additional>=ship)
40      {
41          cout<<"Wrong data entered "<<endl;
42          goto abc;
43      }
44      int c = ship-additional;
45      cout<<"Number of passengers the ship can carry = "<<c<<"\n\n";
46      int capacity[1000],cn=0;
47      int ch;
48      while(1)
49      {
50          ef:
51          cout<<"\nEnter the number of people in the group of tourists : \n"<<endl;
52          cout<<"****Enter 0 to stop entering the data****"<<endl;
53          cin>>ch;
54          if(ch==0)
55              break;
56          if(ch>c)
57              {
58                  cout<<"number of passengers more than ship capacity "<<endl;
59                  cout<<"can't be accomodated "<<endl;
60                  goto ef;
61              }
62
63          capacity[cn]=ch;
64          cn++;
65      }
66      int n = cn;
67      cout << "Number of ships required in : "
68           << nextFit(capacity, n, c);
69      return 0;
70  }
71
```

Logs & others

Code::Blocks | Search results | Cccc | Build log | Build messages | CppCheck | CppCheck messages | Cscope | Debugger | DoxyBlocks | Fortran info | Closed files list | Thread search

OUTPUT


"C:\Users\Debprotim .LAPTOP-POKADHLI\Desktop\C++ CODES\nextfit.exe"
Enter the maximum number of people the ship can carry :
26
Enter the maximum number of crew members aboard the ship :
6
Number of passengers the ship can carry = 20
Enter the number of people in the group of tourists :
****Enter 0 to stop entering the data****
15
Enter the number of people in the group of tourists :
****Enter 0 to stop entering the data****
5
Enter the number of people in the group of tourists :
****Enter 0 to stop entering the data****
10
Enter the number of people in the group of tourists :
****Enter 0 to stop entering the data****
32
number of passengers more than ship capacity
can't be accomodated
Enter the number of people in the group of tourists :
****Enter 0 to stop entering the data****
16
Enter the number of people in the group of tourists :
****Enter 0 to stop entering the data****
20
Enter the number of people in the group of tourists :
****Enter 0 to stop entering the data****
12
Enter the number of people in the group of tourists :
****Enter 0 to stop entering the data****
8
Enter the number of people in the group of tourists :
Type here to search
02:46 PM
30-03-2018


"C:\Users\Debprotim .LAPTOP-POKADHLJ\Desktop\C++ CODES\nextfit.exe"
can't be accomodated
Enter the number of people in the group of tourists :
****Enter 0 to stop entering the data****
16
Enter the number of people in the group of tourists :
****Enter 0 to stop entering the data****
20
Enter the number of people in the group of tourists :
****Enter 0 to stop entering the data****
12
Enter the number of people in the group of tourists :
****Enter 0 to stop entering the data****
8
Enter the number of people in the group of tourists :
****Enter 0 to stop entering the data****
9
Enter the number of people in the group of tourists :
****Enter 0 to stop entering the data****
0
Number of ships required in : 6
Process returned 0 (0x0) execution time : 54.328 s
Press any key to continue.
Type here to search
02:47 PM
30-03-2018

FOR CARGO SHIP WE USED BEST FIT BIN PACKING ALGORITHM :

THE CODE IS AS FOLLOWS

```
// C++ program to find MINIMUM NUMBER OF CARGO SHIPS REQUIRED
// Best fit algorithm.
#include <iostream>
using namespace std;

// Returns number of bins required using best fit
int bestFit(int weight[], int n, int c)
{
    // Initialize result (Count of ships)
    int res = 0;

    // Create an array to store remaining space in ships
    // there can be at most n ships
    int bin_rem[n];

    // Place items one by one
    for (int i=0; i<n; i++)
```

```
{
    // Find the best ship that can accomodate
    // weight[i]
    int j;

    // Initialize minimum space left and index
    // of best ship
    int min = c+1, bi = 0;

    for (j=0; j<res; j++)
    {
        if (bin_rem[j] >= weight[i] &&
                bin_rem[j] - weight[i] < min)
        {
            bi = j;
            min = bin_rem[j] - weight[i];
        }
    }

    // If no ship could accommodate weight[i],
    // create a new ship
```

```
        if (min==c+1)
        {
            bin_rem[res] = c - weight[i];
            res++;
        }
        else // Assign the item to best ship
            bin_rem[bi] -= weight[i];
    }
    return res;
}

// Driver program
int main()
{
    int ship,oil,crew,additional;
    abc:
    cout<<"Enter the net weight the ship can carry : "<<endl;
    cin>>ship;
    cout<<"Enter the weight of fuel that is to be filled in the ship : "<<endl;
    cin>>oil;
```

```
    cout<<"Enter the maximum weight of the crew members aboard the ship : "<<endl;
    cin>>crew;
    additional=oil+crew;
    if(additional>=ship)
    {
        cout<<"Wrong data entered "<<endl;
        goto abc;
    }
    int c = ship-additional;
    cout<<"Effective cargo that a ship can carry = "<<c<<"\n\n";
    int weight[1000],cn=0;
    int ch;
    while(1)
    {
        ef:
        cout<<"\nEnter the weight : \n"<<endl;
        cout<<"****Enter 0 to stop entering the data****"<<endl;
        cin>>ch;
        if(ch==0)
```

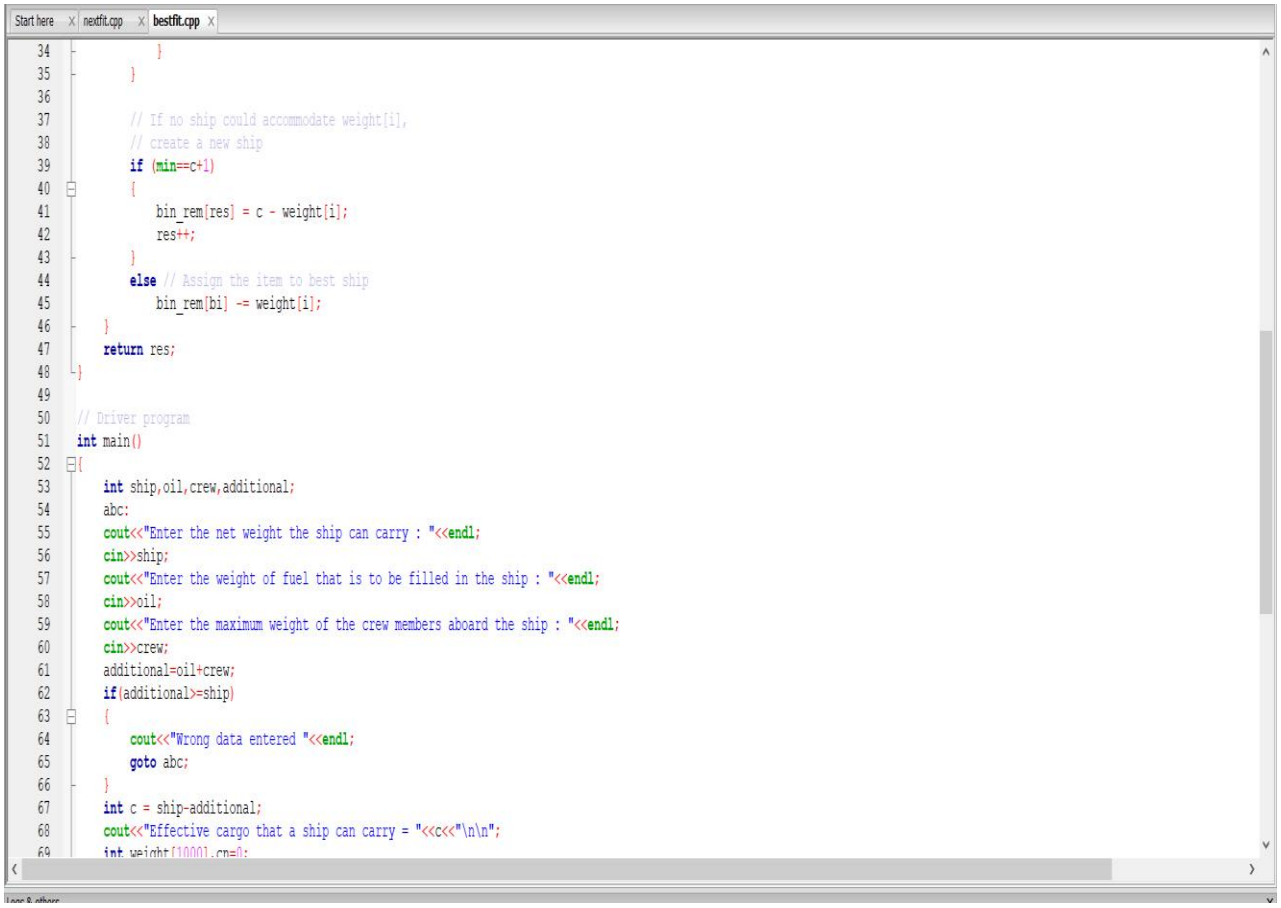
```
            break;
        if(ch>c)
            {
                cout<<"too much weight for ship to carry
"<<endl;
                cout<<"can't be shipped "<<endl;
                goto ef;
            }

        weight[cn]=ch;
        cn++;
    }
    int n = cn;
    cout << "Number of ships required in Best Fit : "
         << bestFit(weight, n, c);
    return 0;
}
```

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

Start here | nextfit.cpp | bestfit.cpp

```cpp
// C++ program to find MINIMUM NUMBER OF CARGO SHIPS REQUIRED
// Best fit algorithm.
#include <iostream>
using namespace std;

// Returns number of bins required using best fit
int bestFit(int weight[], int n, int c)
{
    // Initialize result (Count of ships)
    int res = 0;

    // Create an array to store remaining space in ships
    // there can be at most n ships
    int bin_rem[n];

    // Place items one by one
    for (int i=0; i<n; i++)
    {
        // Find the best ship that can accomodate
        // weight[i]
        int j;

        // Initialize minimum space left and index
        // of best ship
        int min = c+1, bi = 0;

        for (j=0; j<res; j++)
        {
            if (bin_rem[j] >= weight[i] &&
                    bin_rem[j] - weight[i] < min)
            {
                bi = j;
                min = bin_rem[j] - weight[i];
            }
        }
```

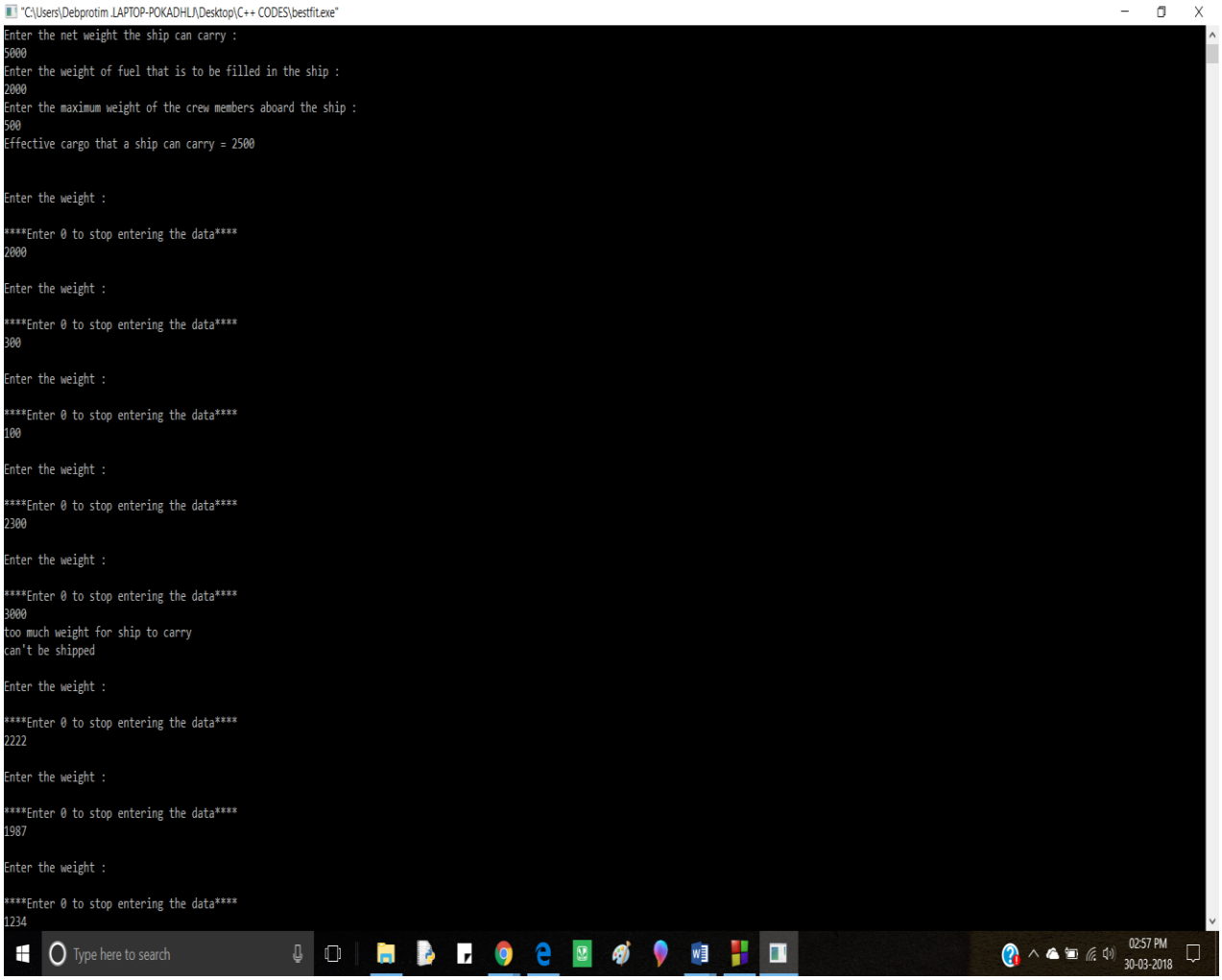
Logs & others

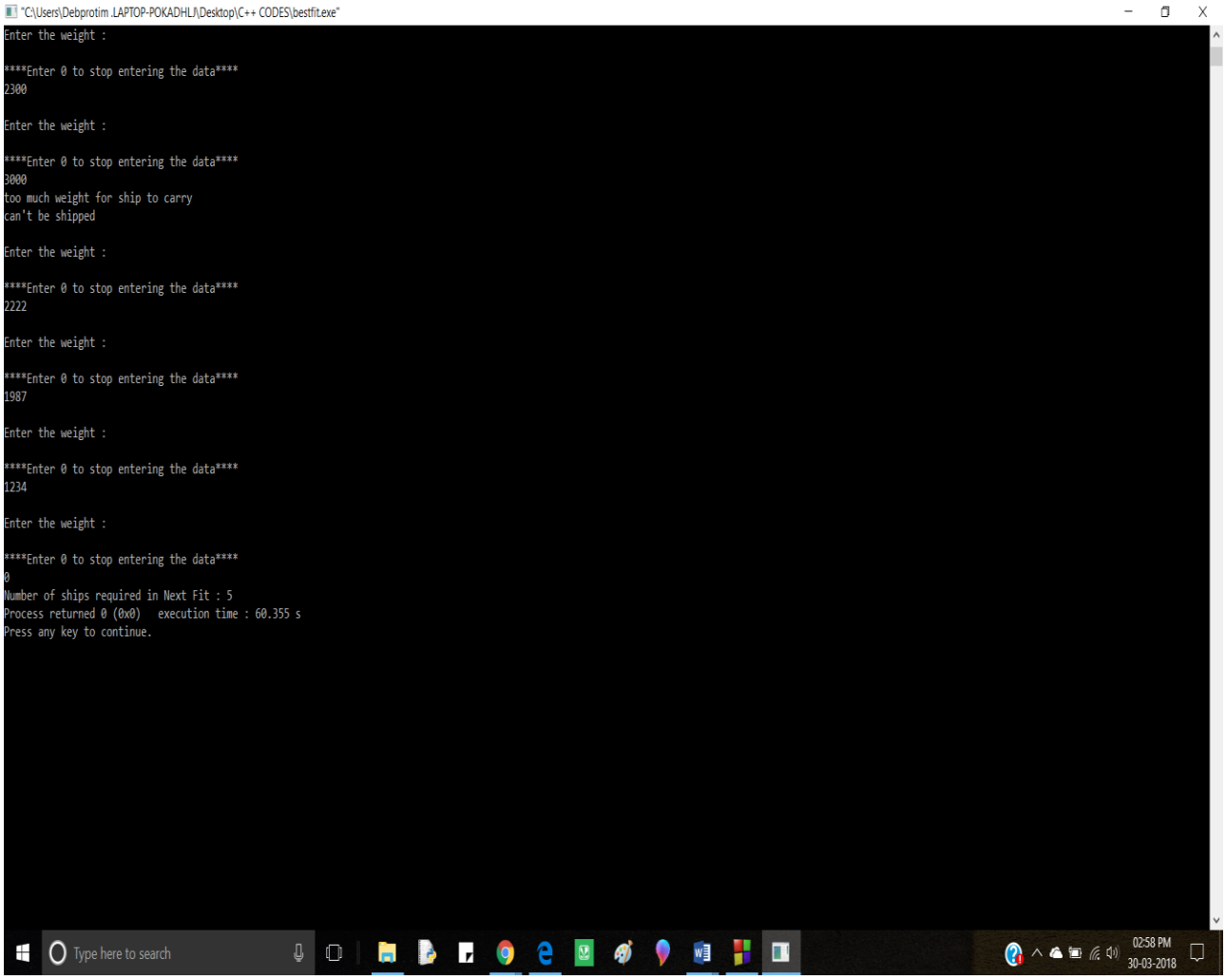

Start here
nextfit.cpp
bestfit.cpp
}
}
// If no ship could accommodate weight[i],
// create a new ship
if (min==c+1)
{
bin_rem[res] = c - weight[i];
res++;
}
else // Assign the item to best ship
bin_rem[bi] -= weight[i];
}
return res;
}
// Driver program
int main()
{
int ship,oil,crew,additional;
abc:
cout<<"Enter the net weight the ship can carry : "<<endl;
cin>>ship;
cout<<"Enter the weight of fuel that is to be filled in the ship : "<<endl;
cin>>oil;
cout<<"Enter the maximum weight of the crew members aboard the ship : "<<endl;
cin>>crew;
additional=oil+crew;
if(additional>=ship)
{
cout<<"Wrong data entered "<<endl;
goto abc;
}
int c = ship-additional;
cout<<"Effective cargo that a ship can carry = "<<c<<"\n\n";
int weight[10001],cn=0;

Start here | nextfit.cpp | bestfit.cpp

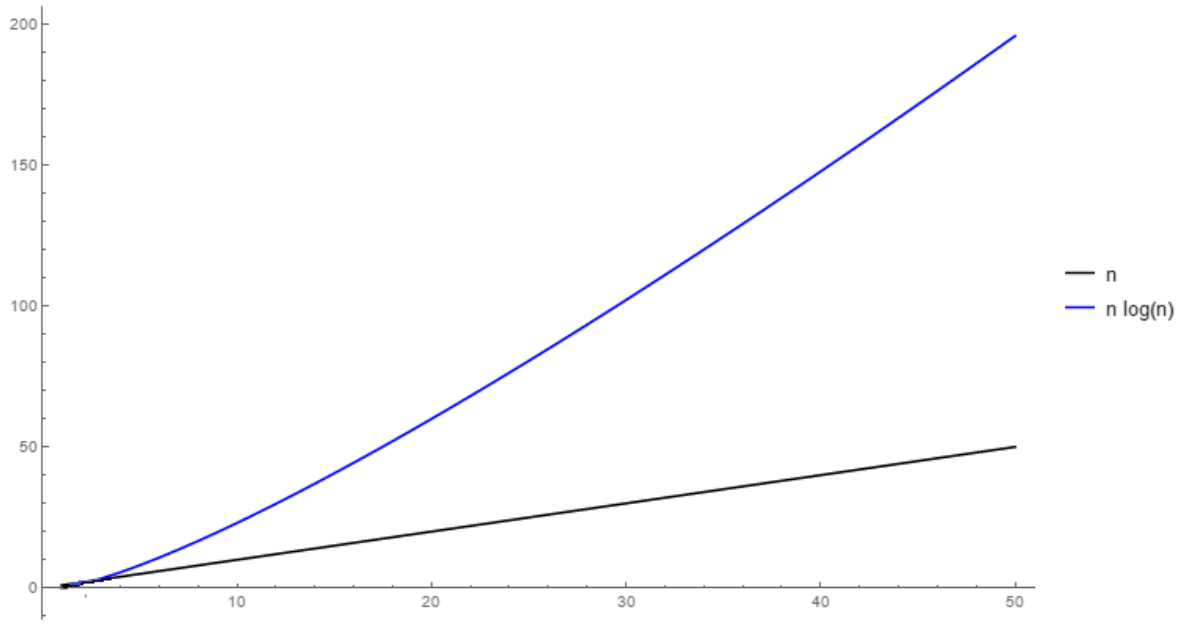
```
61      additional=oil+crew;
62      if(additional>=ship)
63      {
64          cout<<"Wrong data entered "<<endl;
65          goto abc;
66      }
67      int c = ship-additional;
68      cout<<"Effective cargo that a ship can carry = "<<c<<"\n\n";
69      int weight[1000],cn=0;
70      int ch;
71      while(1)
72      {
73          ef:
74          cout<<"\nEnter the weight : \n"<<endl;
75          cout<<"*****Enter 0 to stop entering the data****"<<endl;
76          cin>>ch;
77          if(ch==0)
78              break;
79          if(ch>c)
80              {
81                  cout<<"too much weight for ship to carry "<<endl;
82                  cout<<"can't be shipped "<<endl;
83                  goto ef;
84              }
85
86          weight[cn]=ch;
87          cn++;
88      }
89      int n = cn;
90      cout << "Number of ships required in Next Fit : "
91           << bestFit(weight, n, c);
92      return 0;
93  }
94
95
```

OUTPUT


"C:\Users\Debprotim .LAPTOP-POKADHLJ\Desktop\C++ CODES\bestfit.exe"
Enter the net weight the ship can carry :
5000
Enter the weight of fuel that is to be filled in the ship :
2000
Enter the maximum weight of the crew members aboard the ship :
500
Effective cargo that a ship can carry = 2500
Enter the weight :
****Enter 0 to stop entering the data****
2000
Enter the weight :
****Enter 0 to stop entering the data****
300
Enter the weight :
****Enter 0 to stop entering the data****
100
Enter the weight :
****Enter 0 to stop entering the data****
2300
Enter the weight :
****Enter 0 to stop entering the data****
3000
too much weight for ship to carry
can't be shipped
Enter the weight :
****Enter 0 to stop entering the data****
2222
Enter the weight :
****Enter 0 to stop entering the data****
1987
Enter the weight :
****Enter 0 to stop entering the data****
1234
Type here to search
02:57 PM
30-03-2018


"C:\Users\Debprotim .LAPTOP-POKADHLJ\Desktop\C++ CODES\bestfit.exe"
Enter the weight :
****Enter 0 to stop entering the data****
2300
Enter the weight :
****Enter 0 to stop entering the data****
3000
too much weight for ship to carry
can't be shipped
Enter the weight :
****Enter 0 to stop entering the data****
2222
Enter the weight :
****Enter 0 to stop entering the data****
1987
Enter the weight :
****Enter 0 to stop entering the data****
1234
Enter the weight :
****Enter 0 to stop entering the data****
0
Number of ships required in Next Fit : 5
Process returned 0 (0x0) execution time : 60.355 s
Press any key to continue.
Type here to search
02:58 PM
30-03-2018

ANALYSIS AND INFERENCES USING GRAPH



Here, the linear graph (BLACK LINE) represents O(n) which is time complexity of NEXT FIT ALGORITHM

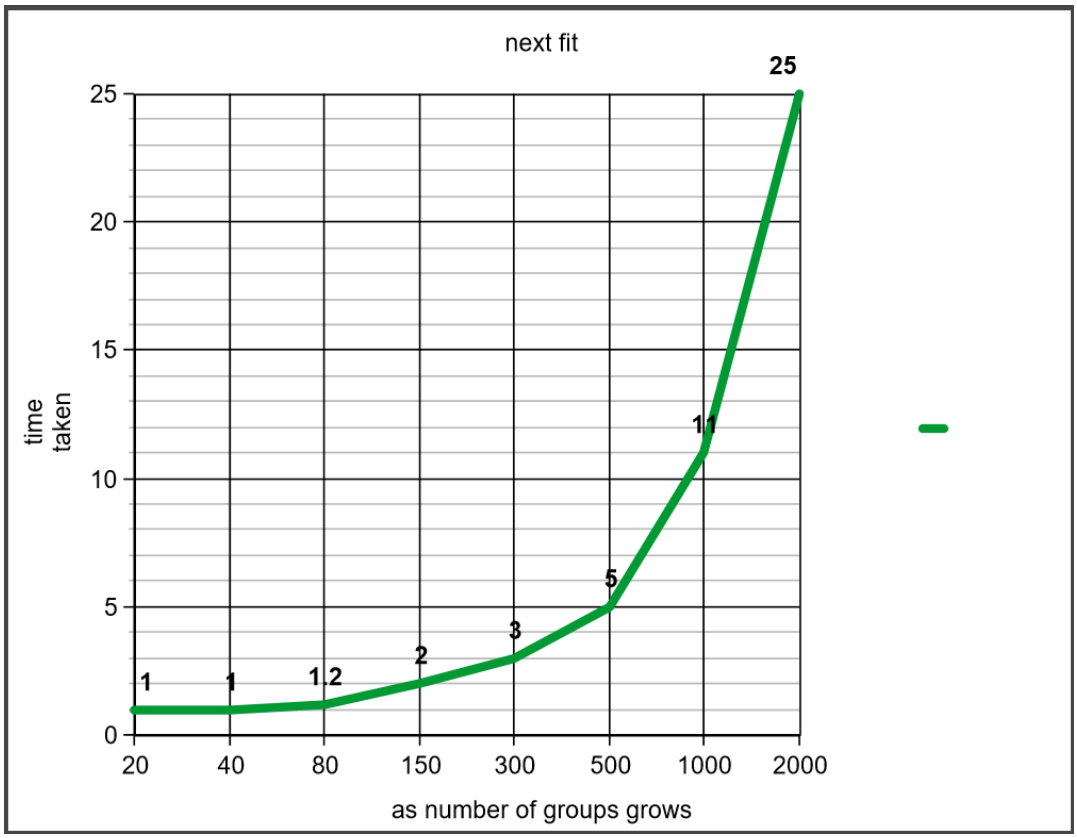
Next Fit is a simple algorithm. It requires only O(n) time and O(1) extra space to process n items.

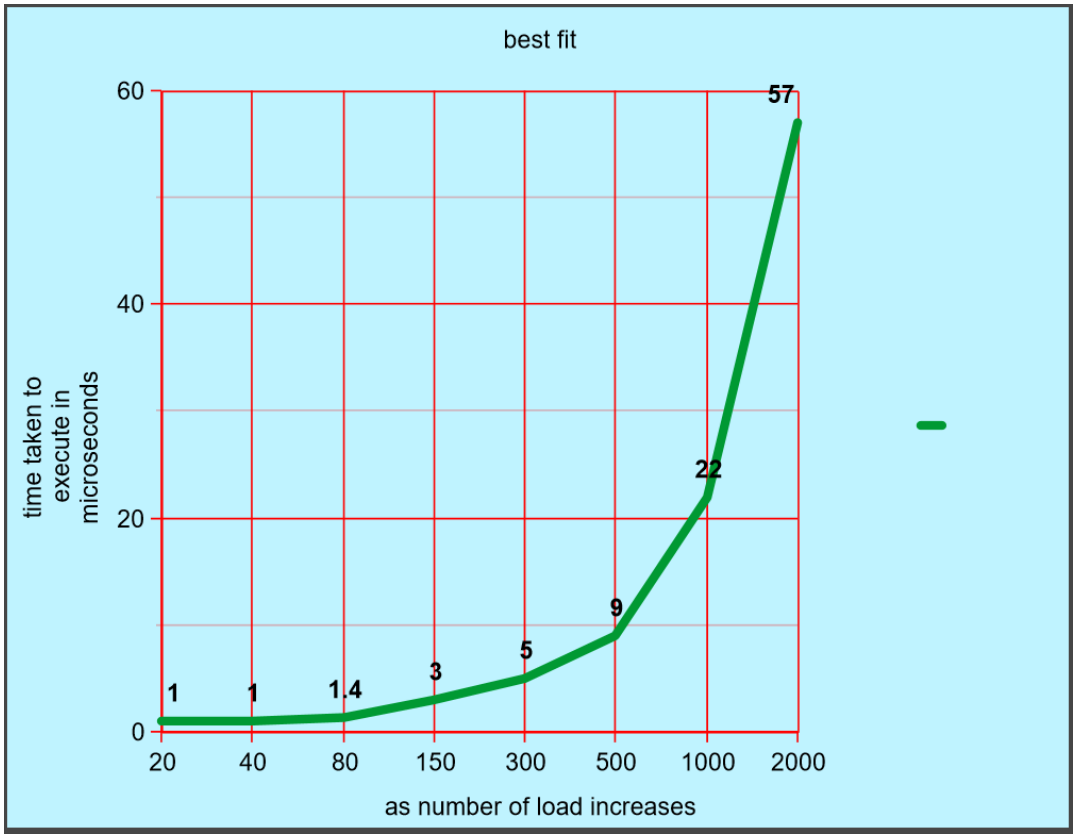
Next Fit is to approximate, i.e., the number of bins used by this algorithm is bounded by twice of optimal. Consider any two adjacent bins. The sum of items in these two bins must be > c; otherwise, NextFit would have put all the items of second bin into the first. The same holds for all other bins. Thus, at most half the space is wasted, and so Next Fit uses at most 2M bins if M is optimal. But, here time complexity is very less.

The other curvilinear graph(BLUE COLOUR) represent O(nlogn) which is time complexity of BEST FIT ALGORITHM.

If M is the optimal number of bins, then Best Fit never uses more than 1.7M bins. Its more accurate then Next fit. But, here time complexity is more.

GRAPHS :


next fit
time taken
as number of groups grows
0
5
10
15
20
25
20
40
80
150
300
500
1000
2000
1
1
1.2
2
3
5
11
25


best fit
time taken to execute in microseconds
as number of load increases
0
20
40
60
20
40
80
150
300
500
1000
2000
1
1
1.4
3
5
9
22
57

CONCLUSION

In bin packing, the time complexity of next-fit algorithm is less. This is the reason that we applied this algorithm to the passenger ferry boat program because in passenger ferry boat business, any delay is not acceptable as many tourists are to be accommodated in many boats. If delay occurs, that would result in inconvenience to the tourists and many tourists would be left out of using passenger ferry boats which would directly affect the profits of these ferry boat companies to decrease. However, the space used by next-fit algorithm is not minimal, but it is best for ferry boat program as their priority is less time without any delays.

Whereas, in best-fit algorithm the space used is minimal. This is the reason we applied this algorithm to cargo shipping program because these shipping cost very much to big companies and less space is their priority because if more materials are shipped in less space, the profits of these companies would increase heavily. However, the time complexity of this best-fit algorithm is more than next-fit algorithm but it is best for cargo shipping program as priority for these companies is using minimal space rather than doing this is in minimal time to ship their material to increase their profit margins.

REFERENCES

1. Algorithms for Two-Dimensional Bin Packing and Assignment Problems by Andrea Lodi, Prof. Giovanni Marro, Prof. Silvano Martello .

2. www.sanfoundry.com .