Justin Phillips

Hw 8 CS325 spring 19

Problem 1:(20 pts)

In the bin packing problem, items of different weights (or sizes) must be packed into a finite number of bins each with the capacity C in a way that minimizes the number of bins used. The decision version of the bin packing problem (deciding if objects will fit into <= k bins) is NP-complete. There is no known polynomial time algorithm to solve the optimization version of the bin packing problem. In this homework you will be examining three greedy approximation algorithms to solve the bin packing problem.

• First -Fit: Put each item as you come to it into the first (earliest opened) bin into which it fits. If
there is no available bin then open a new bin.

•First-Fit-Decreasing: First sort the items in decreasing order by size, then u se First-Fit on the resulting list.

•Best Fit:Place the items in the order in which they arrive. Place the next item into the bin which will leave the least room left over after the item is placed in the bin. If it does not fit in any bin, start a new bin.

a) Give pseudo code and the running time for each of the approximation algorithm

-FF(items, weight, size) // first fit pseudo

make a list to hold the bin list

run a for loop for the items in the list

if we got space available in the bin pace the item

else place item in a new bin

the approximate running time is a worst case O(n^2)

-to run the FFD we need to sort the list 1st then we can run the first fit function

FFD(items, weight, size) // pseudo first-fit decreasing

make a list to hold the bin list

run a for loop for the items in the list

if we got space available in the bin pace the item

else place item in a new bin

the approximate running time is a worst case O(n^2), also to factor in the algorithm would be which sort algorithm we chose to sort our list.

-BF(items, weight, size) // best fit pseudo

make a list to hold the bin list

run a for loop for the items in the list

if there is space in the bin pace it in it, if there are more then one bin open compare the bins and see which bin would have the least amount of space left in it

else no bins have space place in a new one

the approximate running time is a worst case O(n^2)

c) Randomly generate at least 20 bin packing instances of varying sizes(number of items).Submit a description of how the inputs were generated not the code used to produce the random inputs.

Test 1:
Items 91:
 First Fit: 52
 Best Fit: 52
 First Fit Decreasing: 50
Test 2:
Items 40:
 First Fit: 24
 Best Fit: 24
 First Fit Decreasing: 23
Test 3:
Items 90:
 First Fit: 53
 Best Fit: 52
 First Fit Decreasing: 51
Test 4:
Items 78:
 First Fit: 44
 Best Fit: 44
 First Fit Decreasing: 42
Test 5:
Items 12:
 First Fit: 5
 Best Fit: 5
 First Fit Decreasing: 5
Test 6:
Items 36:
 First Fit: 26
 Best Fit: 26

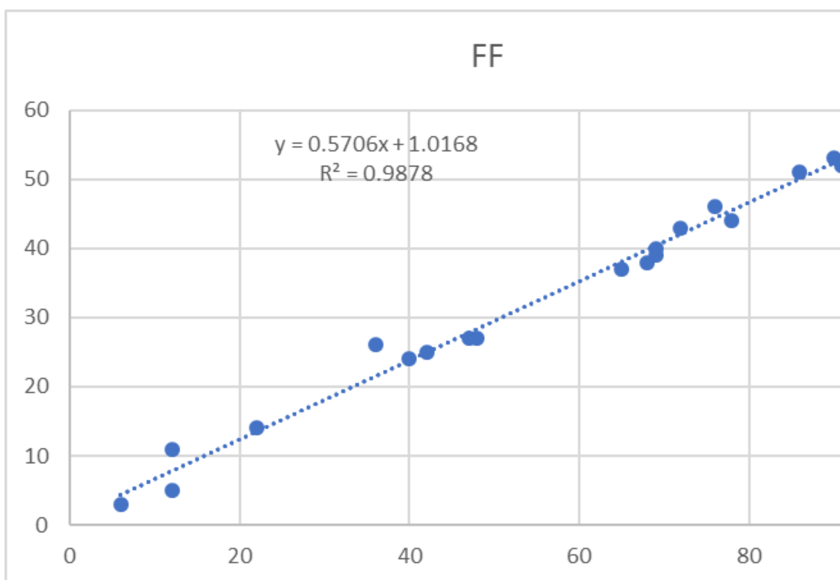
First Fit Decreasing: 25
Test 7:
Items 48:
First Fit: 27
Best Fit: 27
First Fit Decreasing: 27
Test 8:
Items 72:
First Fit: 43
Best Fit: 43
First Fit Decreasing: 42
Test 9:
Items 47:
First Fit: 27
Best Fit: 27
First Fit Decreasing: 26
Test 10:
Items 6:
First Fit: 3
Best Fit: 3
First Fit Decreasing: 3
Test 11:
Items 92:
First Fit: 55
Best Fit: 54
First Fit Decreasing: 54
Test 12:
Items 69:
First Fit: 39
Best Fit: 39
First Fit Decreasing: 37
Test 13:
Items 86:
First Fit: 51
Best Fit: 51
First Fit Decreasing: 51
Test 14:
Items 65:
First Fit: 37
Best Fit: 36
First Fit Decreasing: 35
Test 15:
Items 76:
First Fit: 46
Best Fit: 46
First Fit Decreasing: 45

Test 16:
Items 22:
First Fit: 14
Best Fit: 13
First Fit Decreasing: 13
Test 17:
Items 12:
First Fit: 11
Best Fit: 11
First Fit Decreasing: 10
Test 18:
Items 42:
First Fit: 25
Best Fit: 25
First Fit Decreasing: 23
Test 19:
Items 68:
First Fit: 38
Best Fit: 38
First Fit Decreasing: 36
Test 20:
Items 69:
First Fit: 40
Best Fit: 39
First Fit Decreasing: 38

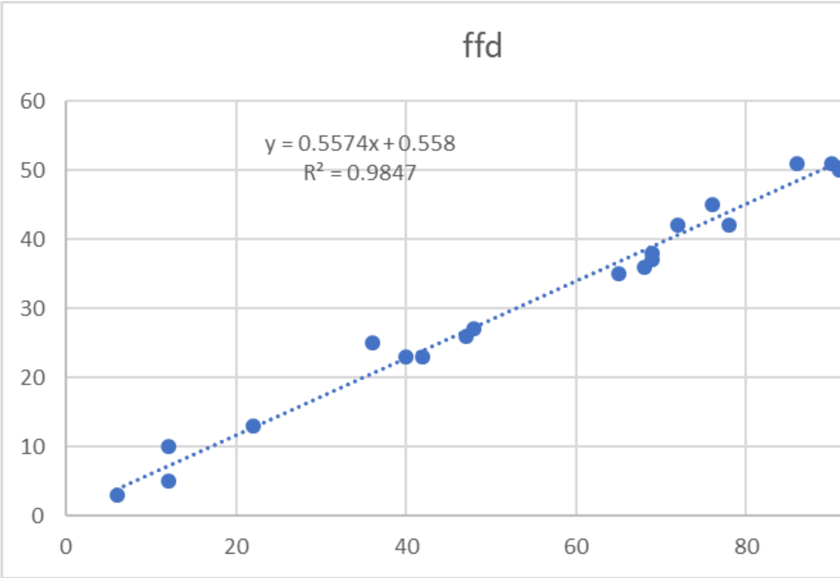
i.Create a graph or chart that shows the number of bins used by each algorithm with instances of different sizes. Which algorithm performs better if the metric is number of bins?
as the number of bins increases the ffd algorithm performs the best.

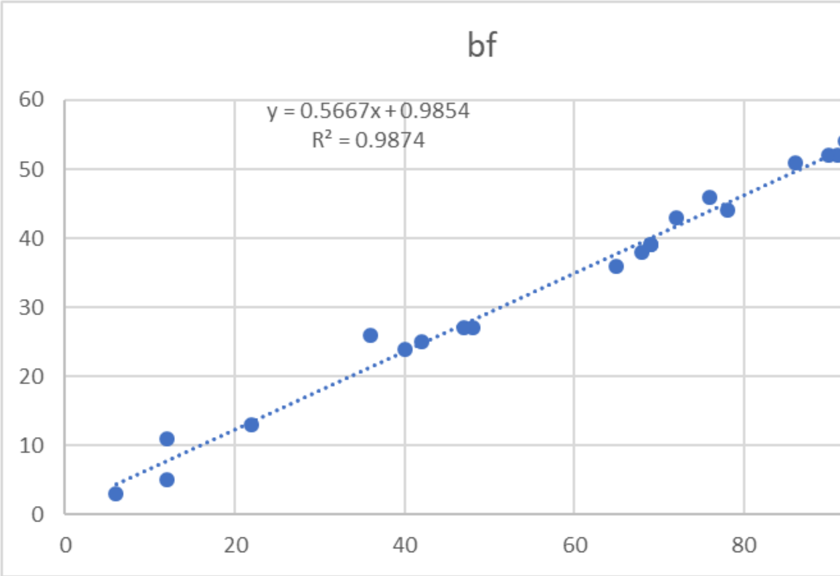
ii.Create a graph or chart that shows the running time each algorithm as a function of numbers of items. Which algorithm performs better if the metric is running time?
the fastest running time for the algorithms is ff
x = items, y=bins

items	FF	ffd	bf
91	52	50	52
40	24	23	24
90	53	51	52
78	44	42	44
12	5	5	5
36	26	25	26
48	27	27	27
72	43	42	43
47	27	26	27



6	3	3	3
92	55	54	54
69	39	37	39
86	51	51	51
65	37	35	36
76	46	45	46
22	14	13	13
12	11	10	11
42	25	23	25
68	38	36	38
69	40	38	39





Problem 2:(10 pts)

An exact solution to the bin packing optimization problem can be found using 0-1 integer programming (IP)see the format on the Wikipedia page. Write an integer program for each of the following instances of bin packing and solve with the software of your choice. Submit a copy of the code and interpret the results.

a) Six items S= { 4, 4, 4, 6, 6, 6} and bin capacity of 10

```
MIN y1+y2+y3+y4+y5+y6
ST
y1+y2+y3+y4+y5+y6 > 0

4x11 + 4x12 + 4x13 + 6x14 + 6x15 + 6x16 - 10y1 <= 0
4x21 + 4x22 + 4x23 + 6x24 + 6x25 + 6x26 - 10y2 <= 0
4x31 + 4x32 + 4x33 + 6x34 + 6x35 + 6x36 - 10y3 <= 0
4x41 + 4x42 + 4x43 + 6x44 + 6x45 + 6x46 - 10y4 <= 0
4x51 + 4x52 + 4x53 + 6x54 + 6x55 + 6x56 - 10y5 <= 0
4x61 + 4x62 + 4x63 + 6x64 + 6x65 + 6x66 - 10y6 <= 0

x11 + x21 + x31 + x41 + x51 + x61 = 1
x12 + x22 + x32 + x42 + x52 + x62 = 1
x13 + x23 + x33 + x43 + x53 + x63 = 1
x14 + x24 + x34 + x44 + x54 + x64 = 1
x15 + x25 + x35 + x45 + x55 + x65 = 1
x16 + x26 + x36 + x46 + x56 + x66 = 1
END
INT y1
INT y2
INT y3
INT y4
INT y5
INT y6
INT x11
INT x12
INT x13
INT x14
INT x15
INT x16
INT x21
INT x22
INT x23
INT x24
INT x25
INT x26
INT x31
INT x32
INT x33
INT x34
```

INT x35
INT x36
INT x41
INT x42
INT x43
INT x44
INT x45
INT x46
INT x51
INT x52
INT x53
INT x54
INT x55
INT x56
INT x61
INT x62
INT x63
INT x64
INT x65
INT x66

```
NEW INTEGER SOLUTION OF    3.00000000     AT BRANCH      0 PIVOT      26
 RE-INSTALLING BEST SOLUTION...

        OBJECTIVE FUNCTION VALUE

        1)      3.000000

  VARIABLE        VALUE          REDUCED COST
        Y1         0.000000          1.000000
        Y2         1.000000          1.000000
        Y3         0.000000          1.000000
        Y4         1.000000          1.000000
        Y5         1.000000          1.000000
        Y6         0.000000          1.000000
       X11         0.000000          0.000000
       X12         0.000000          0.000000
       X13         0.000000          0.000000
       X14         0.000000          0.000000
       X15         0.000000          0.000000
       X16         0.000000          0.000000
       X21         1.000000          0.000000
       X22         0.000000          0.000000
       X23         0.000000          0.000000
       X24         0.000000          0.000000
       X25         1.000000          0.000000
```

X26	0.000000	0.000000
X31	0.000000	0.000000
X32	0.000000	0.000000
X33	0.000000	0.000000
X34	0.000000	0.000000
X35	0.000000	0.000000
X36	0.000000	0.000000
X41	0.000000	0.000000
X42	1.000000	0.000000
X43	0.000000	0.000000
X44	1.000000	0.000000
X45	0.000000	0.000000
X46	0.000000	0.000000
X51	0.000000	0.000000
X52	0.000000	0.000000
X53	1.000000	0.000000
X54	0.000000	0.000000
X55	0.000000	0.000000
X56	1.000000	0.000000
X61	0.000000	0.000000
X62	0.000000	0.000000
X63	0.000000	0.000000
X64	0.000000	0.000000
X65	0.000000	0.000000
X66	0.000000	0.000000

ROW	SLACK OR SURPLUS	DUAL PRICES
2)	3.000000	0.000000
3)	0.000000	0.000000
4)	0.000000	0.000000
5)	0.000000	0.000000
6)	0.000000	0.000000
7)	0.000000	0.000000
8)	0.000000	0.000000
9)	0.000000	0.000000
10)	0.000000	0.000000
11)	0.000000	0.000000
12)	0.000000	0.000000
13)	0.000000	0.000000
14)	0.000000	0.000000

NO. ITERATIONS= 26
BRANCHES= 0 DETERM.= 1.000E 0

the optimal solution for this is 3 bins.

b) Five items S = { 20, 10, 15, 10, 5} and bin capacity of 20

```
MIN y1+y2+y3+y4+y5
ST
y1+y2+y3+y4+y5 > 0

20x11 + 10x12 + 15x13 + 10x14 + 5x15 - 20y1 <= 0
20x21 + 10x22 + 15x23 + 10x24 + 5x25 - 20y2 <= 0
20x31 + 10x32 + 15x33 + 10x34 + 5x35 - 20y3 <= 0
20x41 + 10x42 + 15x43 + 10x44 + 5x45 - 20y4 <= 0
20x51 + 10x52 + 15x53 + 10x54 + 5x55 - 20y5 <= 0

x11 + x21 + x31 + x41 + x51 = 1
x12 + x22 + x32 + x42 + x52 = 1
x13 + x23 + x33 + x43 + x53 = 1
x14 + x24 + x34 + x44 + x54 = 1
x15 + x25 + x35 + x45 + x55 = 1
END
INT y1
INT y2
INT y3
INT y4
INT y5
INT x11
INT x12
INT x13
INT x14
INT x15
INT x21
INT x22
INT x23
INT x24
INT x25
INT x31
INT x32
INT x33
INT x34
INT x35
INT x41
INT x42
INT x43
INT x44
INT x45
INT x51
INT x52
INT x53
```

INT x54
INT x55

LP OPTIMUM FOUND AT STEP 17
OBJECTIVE VALUE = 3.00000000

NEW INTEGER SOLUTION OF 3.00000000 AT BRANCH 0 PIVOT 17
RE-INSTALLING BEST SOLUTION...

OBJECTIVE FUNCTION VALUE

1) 3.000000

VARIABLE	VALUE	REDUCED COST
Y1	1.000000	1.000000
Y2	1.000000	1.000000
Y3	1.000000	1.000000
Y4	0.000000	1.000000
Y5	0.000000	1.000000
X11	0.000000	0.000000
X12	0.000000	0.000000
X13	1.000000	0.000000
X14	0.000000	0.000000
X15	1.000000	0.000000
X21	1.000000	0.000000
X22	0.000000	0.000000
X23	0.000000	0.000000
X24	0.000000	0.000000
X25	0.000000	0.000000
X31	0.000000	0.000000
X32	1.000000	0.000000
X33	0.000000	0.000000
X34	1.000000	0.000000
X35	0.000000	0.000000
X41	0.000000	0.000000
X42	0.000000	0.000000
X43	0.000000	0.000000
X44	0.000000	0.000000
X45	0.000000	0.000000
X51	0.000000	0.000000
X52	0.000000	0.000000
X53	0.000000	0.000000
X54	0.000000	0.000000
X55	0.000000	0.000000

ROW	SLACK OR SURPLUS	DUAL PRICES
2)	3.000000	0.000000
3)	0.000000	0.000000
4)	0.000000	0.000000
5)	0.000000	0.000000
6)	0.000000	0.000000
7)	0.000000	0.000000
8)	0.000000	0.000000
9)	0.000000	0.000000
10)	0.000000	0.000000
11)	0.000000	0.000000
12)	0.000000	0.000000

NO. ITERATIONS= 17
BRANCHES= 0 DETERM.= 1.000E 0

the optimal solution for this is 3 bins.