Quiz 11

Name: Zian Wang

(Test case here)

Program Description:

I implement algorithm 6.2 in the text book to this program. The knapsack and bound functions are the same as the codes in the text book.

I created 1 java class named Priority_queue to define the priority queue in the program, and created another java class named Node to define the nodes in the queue.

Node.java

```
public Node next;
public Node previous;
public int level;
public int profit;
public int weight;
public double bound;
```

These are the member variables of Node.java. Node contains level, profit, weight and bound. The next and previous is the pointer to the next node and previous node in the queue.

Priority_queue.java

The Priority_queue.java has 3 functions: insert, remove and empty.

public void insert(Node a)
Inputs: node a.
Outputs: none.
Function: insert node a into the priority queue.

The queue is arranged from largest to smallest bound value. When insert into a node, the insert function will look for the first node whose bound is smaller than the node a which is waiting to be inserted, and then insert node a before the smaller bound node. If node a is smaller than any node in the queue, node a will be inserted into the tail.

public Node remove()
Inputs: none.
Outputs: the removed node.
Function: remove the first node in the priority queue.

public boolean empty()
Inputs: none.
Outputs: a boolean variable shows whether the priority queue is empty.
Function: check whether the priority queue is empty.

Test cases:

```
"C:\Program Files\Java\jdk-15.0.2\bin\java.exe" "-javaagent:A:\Intell
All the items:
i   pi  wi  pi/wi
1   20  2   10
2   30  5   6
3   35  7   5
4   12  3   4
5   3   1   3
Maximum total weight is: 13

The max profit is: 70
The number of nodes visited before finding the optimal solution: 8

Process finished with exit code 0
```