**Programming Assignment 3**

Job scheduling

104303206黃筱晴

一、解法與實作過程

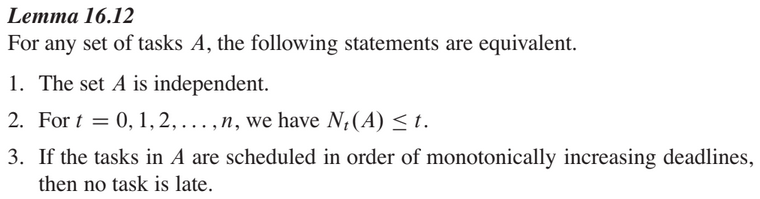
使用貪婪演算法來解決此問題。首先，將所有工作依照截止期限排好，接著將排序過的工作依序加入排程中。每加入一項工作，即檢查排程是否仍為可行序列(所有工作都能在截止期限前完成的)。若是，則繼續加入下一個元素；若非，將排程中工作時間最長的工作刪除，維持其為可行序列。加入所有元素並完成檢查後，排程即為能有最大利潤(完成最多項工作)的最佳序列。

因為不斷遇到尋找最大、最小值的過程，所以使用Priority Queue資料結構來完成演算法。

另外，特別為排程設計了一個schedule類別，其資料成員包含整數 total\_time (總工作時間)，和一個time越大，優先次序則越高的優先佇列J(可行集合)。

借助Priority Queue來完成排序。一開始先將輸入資料推入一個deadline越大，優先次序則越低的優先佇列S中。接著將S中的元素pop取出。deadline最小的工作會越先拿出來。

宣告一排程schedule MaxProfit。呼叫schedule類別的成員函式add\_job，將拿出來的元素加至排程中。add\_job函式內容包含將新元素推入J中；total\_time+=新元素的工作時間；以及檢查是否仍為可行序列。



根據定理，J中的元素集合為可行集合，若且為若J中的工作依截止期限大小的非遞減順序排列之序列必為可行序列。所以現在只需檢查total\_time會不會超過最新推入的元素之deadline即可。如果超過了，則將J中工作時間最長的元素移除(pop掉最優先元素)，並更新total\_time。

最後，S中的元素全部被取出。J中的元素個數就是排程可以獲得的最大利潤。

二、虛擬碼

將所有工作按照截止期限排成序列S

J=空集合;

total\_time=0;

while( S不為空){

將所選的元素加入J中;

total\_time +=所選元素的工作時間

if(total\_time>所選元素的截止期限){

total\_time -= J中最優先元素的工作時間;

J.pop();

}

S.pop();

}

三、複雜度分析

將輸入資料推入S和拿出來-> 花費O(n lg n)

將所選的元素加入J中-> 花費O( lg n)

若不可行，J.pop() -> O( lg n)

以上兩行做了n次共花費O( n lg n)

故整體時間複雜度為O(n lg n)

四、測試資料輸入

執行PA3.exe

