第四次程式作業　資工2A 109502510 張芷烜

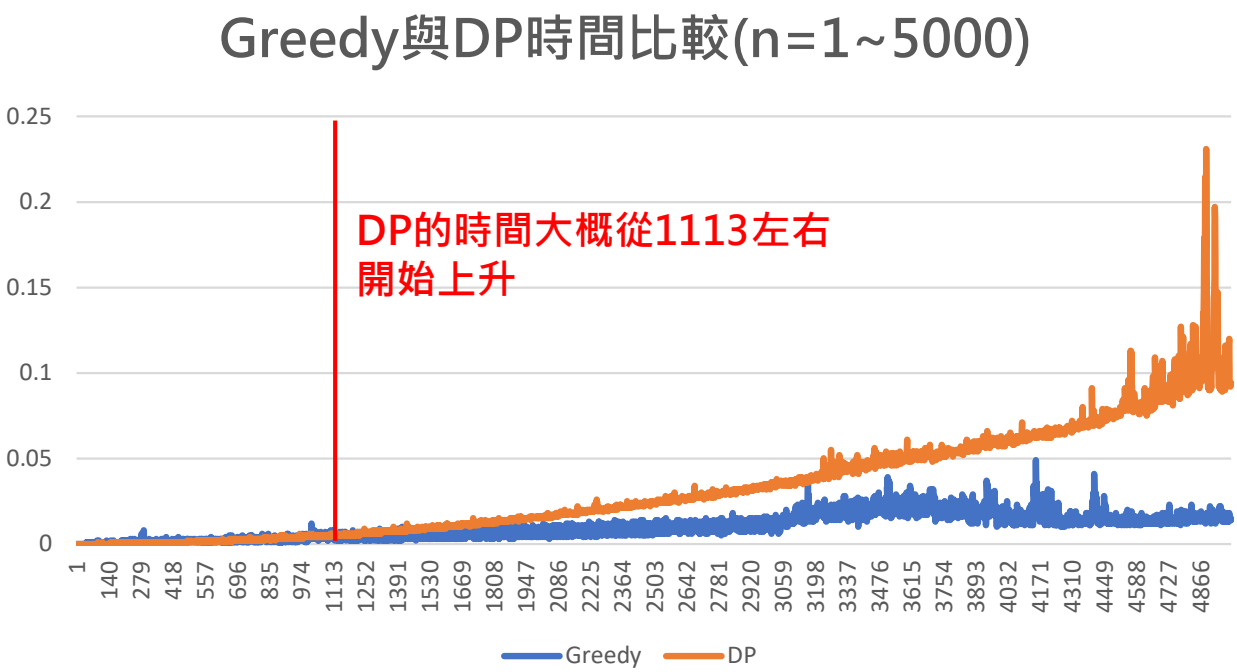
1. Greedy程式碼:

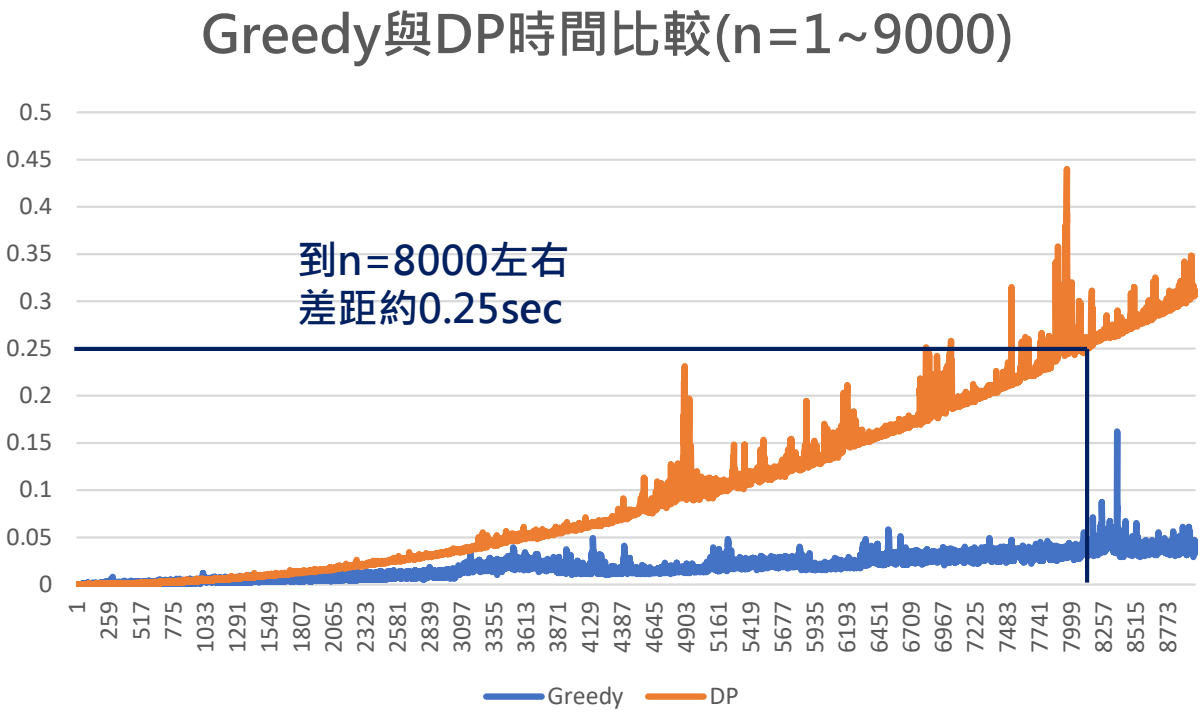
```
int main(){
    double START,END;
    ofstream ofs;
    ofs.open("Greedy.txt");
    for(int n=1;n<=10000;n++){
        for(int i=0;i<n;i++){
            jobs[i].order=i+1;
            jobs[i].time=rand()%100000+1;
            jobs[i].deadline=rand()%100000+1;
        }
        START = clock();
        sort(jobs,jobs+n,cmp);
        int sum=0;

        for(int i=0;i<n;i++){
            ans.push_back(i);
            sum+=jobs[i].time;
            if(sum>jobs[i].deadline){
                sum=sum-FindMAX();  //找到job time最大並移除
            }
        }
        END = clock();
        ofs <<(END - START) / CLOCKS_PER_SEC  << "\n";
        cout<<"n="<<n<<"OK!"<<endl;
        ans.clear();
    }
}
```

2. DP程式碼:與Greedy測量時間方法相同，改變中間實作部分

```
for(int i=0;i<n+1;i++){
    DP[i][0]=0;
    for(int j=1;j<n+1;j++){
        DP[i][j]=INT_MAX;
    }
}
for (int i = 1; i <= n; i++){
    for (int k = 1; k<=i; k++){
        DP[i][k]=DP[i-1][k];
        if(DP[i-1][k-1]!=INT_MAX && DP[i-1][k-1]<=jobs[i-1].deadline-jobs[i-1].time){
            if(DP[i][k]<DP[i-1][k-1]+jobs[i-1].time)
                DP[i][k]=DP[i][k];
            else
                DP[i][k]=DP[i-1][k-1]+jobs[i-1].time;
        }
    }
}
vector<int> ans;
int a=n,b=n;
while(a!=0 && b!=0){
    while(DP[a][b]==INT_MAX){
        b--;
    }
    if(DP[a-1][b-1]+jobs[a-1].time==DP[a][b]){
        ans.push_back(jobs[a-1].order);
        a--;
        b--;
    }
    else{
        a--;
    }
}
```

分析:
將時間計算出來後寫入TXT，再由excel製成圖表如下





結論:

由上圖可以發現當n=1113左右，DP方法開始與Greedy產生差距

到n=8000左右已經有明顯0.25sec左右的差距

註解:DP時間計算有包含找答案