程式作業 3

1. 演算法設計：用了什麼演算法、資料結構

   使用 dynamic program 的演算法，並以 Matrix-Chain Multiplication 尋找最小值為概念進行發想，主要使用資料結構為二維陣列、一維陣列、以及 vector

2. Pseudo code：請勿把程式碼整個貼上來，勿分頁

```
main(){
    while(stringstream >>temp_int) //分割運算元運算子
        num.push_back(temp_int)
        stringstream >>temp_ch
        op.push_back(temp_ch)
    for(int i=0;i<numsize;i++)//最大最小值陣列初始值
        maxx[i][i]=num[i]
        minn[i][i]=num[i]
    for(int x=1;x<=opsize;x++)//max&min dynamic program
        for(int down=0;down<=opsize-x;down++)//下限
            int upp=down+x//上限
            for(int middle=down;middle<upp;middle++){
                switch(op[middle]=='+',' - ',' * ')
                    max{maxx[down][middle]+-*maxx[middle+1][upp]
                        maxx[down][middle]+-*minn[middle+1][upp]
                        minn[down][middle]+-*maxx[middle+1][upp]
                        minn[down][middle]+-*minn[middle+1][upp]}
                    min{maxx[down][middle]+-*maxx[middle+1][upp]
                        maxx[down][middle]+-*minn[middle+1][upp]
                        minn[down][middle]+-*maxx[middle+1][upp]
                        minn[down][middle]+-*minn[middle+1][upp]}
                if(maxnum>maxx[down][upp])
                    maxx[down][upp]=maxnum
                    arg[down][upp]=middle
                if(minnum<minn[down][upp])
                    minn[down][upp]=minnum
                    arg[upp][down]=middle
    findrout(0,opsize) //路徑回推
    parament(0,numsize-1) //括號計算
    for(int i=0;i<numsize;i++) //輸出
        while(xparenth[i]>0)
            cout<<"("
            xparenth[i]--
```

```
            cout<<num[i]
            while(yparenth[i]>0)
                cout<<")"
                yparenth[i]--
            if(i<opsize)
                cout<<op[i]
void findrout(int xline,int yline){
    if(xline!=yline){
            int savemiddle=0
            for(int middle=xline;middle<yline;middle++){
                  if(maxx[xline][yline]來自 min 表 or minn[xline][yline]來自 max 表
                  or 上一層 maxx[xline][yline]來自 min 表){
                arg[xline][savemiddle]=arg[savemiddle][xline]
                arg[savemiddle+1][yline]=arg[yline][savemiddle+1]
            findrout(xline,savemiddle)
            findrout(savemiddle+1,yline)
void parament(int a,int b){
    if(a!=b)
           xparenth[a]++
           yparenth[b]++
           int number = arg[a][b]
           parament(a,number)
           parament(number+1,b)
```

3. 時間複雜度分析：請分析你們所使用的演算法時間複雜度

   主要工作區使用三層迴圈時間複雜度為 O(n^3)，同 Matrix-Chain Multiplication

4. 實驗分析 ： 畫出 n 與時間關係 t 的折線圖