PA2 Report

電機三 梁正 B08901169

Maximum Planar Subset (Bottom-up Algorithm)

```
//bottom-up
int findSubset(int** MPS, int _2N){
    for (int i=0; i<_2N; i++){
        MPS[i][i] = 0;
    }

    int i,j,k;
    for (int l=1; l<_2N; l++){
        for (i=0; i<_2N-l; i++){
            j = i + l;
            k = Chords[j];
            if (k < i || k > j){
                MPS[i][j] = MPS[i][j-1];
            }
            else if (k > i && k < j){
                if (MPS[i][j-1] >= 1 + MPS[i][k-1] + MPS[k+1][j-1]){
                    MPS[i][j] = MPS[i][j-1];
                }else{
                    MPS[i][j] = 1 + MPS[i][k-1] + MPS[k+1][j-1];
                }
            }
            else if (k == i){
                if (l == 1){
                    MPS[i][j] = 1;
                }else{
                    MPS[i][j] = 1 + MPS[i+1][j-1];
                }
            }
        }
    }
    return MPS[0][_2N-1];
}
```

In this PA, I use a vector to store the input of chords, and a 2-D array to store maximum planar subset.

```
vector<int> Chords(180000);
```

```
//initialize M table
int** MPS = new int*[_2N];
for (int i=0; i<_2N; i++){
    MPS[i] = new int[_2N];
}
```

I was trying to implement top-down algorithm but failed to achieve it.