PA2 Report

- b10901059 林咏毅

- **Optimal substructure**
  - Reference: **https://ieeexplore.ieee.org/document/1270250**

    (https://ieeexplore.ieee.org/document/1270250)

    IF $i \le k \le j-1$
    and $|\mathrm{MIS}(i,k-1)| + 1 + |\mathrm{MIS}(k+1,j-1)| > |\mathrm{MIS}(i,j-1)|$
    THEN $\mathrm{MIS}(i,j) \leftarrow \mathrm{MIS}(i,k-1) \cup \{v_{kj}\} \cup \mathrm{MIS}(k+1,j-1)$
    ELSE $\mathrm{MIS}(i,j) \leftarrow \mathrm{MIS}(i,j-1)$
  - Implement with recursion and dynamic programming

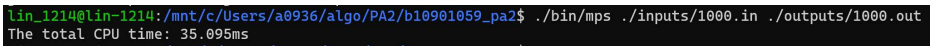
- **Data structures used**

```
// save all the chords from input
// chords[head] = tail
// chords[tail] = head
vector<int> chords(pivot_num, 0);

// save chords used in answer
vector<int> ans_chords;

// up-right half: number of available chords in partition chord_cnt[i][j]
// bottom-left half: record the chord we want
vector<vector<int>> chord_cnt(pivot_num, vector<int> (pivot_num, 0));
for (int i=0; i<pivot_num; i++) {
    for (int j=0; j<pivot_num; j++) {
        if (j < i) {
            chord_cnt[i][j] = -1;
        }
    }
}
```

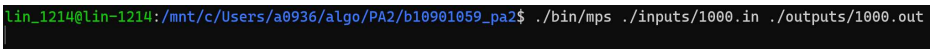
- **Observation**
  - In this assignment, we can see the huge **difference in execution time between bottom-up and top-down method**. Since I first try to implement with bottom-up method, I found it consumes more time than top-down method does, because it computes every result no matter it's useful or not.
  - With dynamic programming, it can reduce a lot of time calculating the the recursion function everytime, below is the comparison using 1000 as

the input:

DP:



non-DP:



(seems it never ends...)