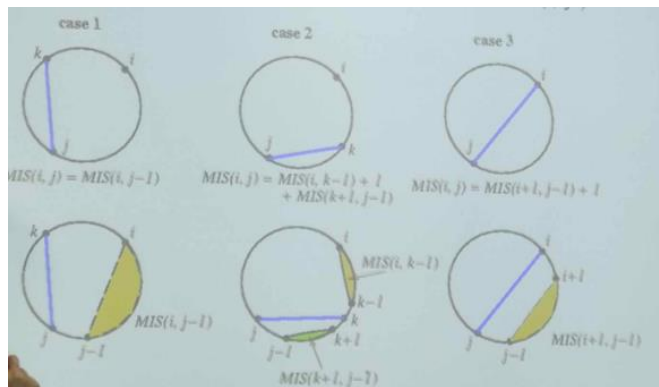


## Data structure:

我用了一個 2 維 **vector**（也就是 **DP**）來存  $i, j$  所對應的最佳組數，所以，當我把表格填完， $(0, 2n-1)$  這一格應該要有我的答案。

除此之外，為了找有哪些組 **chord**，我用了一個一維 **vector** (**chord\_ans**)來記我用了哪些 **chord**。透過查已經建好的 **DP** 的表以及底下三個 **case** 的判斷來一路找回去。最後，我也用了一個一維 **vector** (**chord**) 來快速對應 **chord** 的 **start\_points** 和 **end\_points**，例如：3-5 的 **chord**，我令  $\text{chord}[3] = 5, \text{chord}[5] = 3$ ，我可以透過這種方式來提高效率。

## Subproblems and recurrence formula :



case 2 的話,  $MIS(i, k-1) + 1 + MIS(k+1, j-1)$  還必需要跟  $MIS(i, j-1)$  比大小，找比較大的那個。

## Time and space complexity analysis :

空間大小的部份我必須開上述三個 **vector**，一個二維兩個一維，所以  $\Theta(n^2)$  而時間的部份則因為我不用跑完整個 **DP** 的格子（用的是 **top-down**），所以  $O(n^2)$ 。

## Verification of your analyses :

針對 12 的 **case**，我有親自手畫過確認一樣。其他更大的 **case** 有和其他用不同方法寫的同學比較過，確認沒有太大問題。