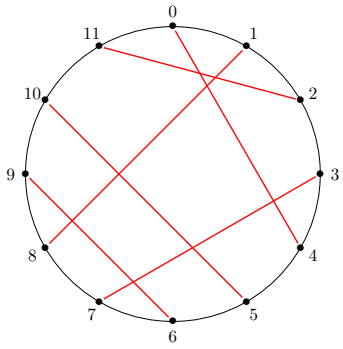
Donat un conjunt de n cordes en el cercle unitat diem que un subconjunt de cordes es *viable* si no hi han dues cordes que es tallen. Volem trobar un subconjunt viable amb mida màxima.

Per resoldre el problema assumim que mai dues cordes tenen un extrem en comú. Per això podem enumerar els extrems de les n cordes de 0 a $2n-1$ seguint el sentit de les agulles del rellotge. Aleshores, l'entrada del problema consisteix en una seqüència de n parelles dels nombres $0, \ldots 2n-1$ on cada i, $0 \leq i \leq 2n-1$, apareix exactament en una parella. La parella (i, j) representa la corda amb extrems i i j. A la figura següent teniu un exemple de instancia amb 6 cordes:



l'entrada corresponent és $(0,4),(1,8)(11,2),(3,7),(5,10),(9,6)$.

Per $0 \leq i < j \leq 2n-1$, definim $T(i,j)$ com la mida del subconjunt viable més gran que es pot formar amb el conjunt de les cordes (a,b) tals que $i \leq a, b \leq j$.

1. Per $0 \leq i < j \leq 2n-1$, proporcioneu una recurrència que permeti calcular $T(i,j)$.
2. Proporcioneu un algorisme que, donat un conjunt de cordes en el cercle unitat, obtingui un conjunt viable amb mida màxima en temps polinòmic.

Una solució:

Considerem la corda (hi ha exactament una) que té un extrem a j. Tenim dos casos:

a) Si l'altre extrem, $a(j)$, està entre i i $j-1$ (inclosos) llavors aquesta corda pot formar part del subconjunt de cordes viables amb extrems (a,b) i $i \leq a \leq b \leq j$, és a dir, contribuir a $T(i,j)$. Aleshores

$$T(i,j) = \max\{T(i,j-1), \quad 1 + T(i,a(j)-1) + T(a(j)+1, j-1),$$

és a dir, serà el màxim entre:

a.1) afegir la corda $(a(j),j)$ al conjunt i optimitzar la resta de les cordes triades entre totes les que tenen extrems entre i i $a(j)-1$ i les que tenen extrems entre $a(j)+1$ i $j-1$, perquè no poden tallar a la corda $(a(j),j)$, i

a.2) no afegir la corda $(a(j),j)$ al conjunt i llavors optimitzar l'elecció amb cordes els extrems de les quals estan en el rang $[i..j-1]$.

b) En canvi, si l'extrem $a(j)$ està fora del rang $[i..j-1]$ llavors aquesta corda **no** pot formar part de cap subconjunt viable i per tant $T(i,j) = T(i,j-1)$.

L'objectiu del problema és el valor $T(0, 2n-1)$. Per calcular-lo segons se'ns demana a l'apartat (b) apliquem tècniques estàndard de PD. Una matriu $2n \times 2n$ guarda els valors $T_{ij} := T(i,j)$. De fet només necessitem guardar la triangular superior, doncs $T_{ij} = 0$ si $i \geq j$. Farem un preprocés que recorre el conjunt de les cordes (u, v) i fixem $a[u] := v$ i $a[v] := u$. Tot això té cost $\Theta(n)$.

Aleshores per omplir cada $T_{i,j}$ amb $i < j$ podem fer-ho amb cost $\Theta(1)$, una vegada determinem si $k = a[j]$ està dins o fora del rang $[i..j-1]$. Només necessitarem accedir a $T_{i,j-1}$ o $T_{i,k-1}$ i $T_{k+1,j-1}$. Fixeu-vos que això exigeix que en el moment que omplim $T_{i,j}$ estiguin ja emplenades totes les entrades de la fila i amb una columna menor que j **y** totes les files més grans que i (almenys fins a la columna j). És per això que la matriu T s'ha d'emplenar de baix ($i = 2n-2$) a dalt ($i = 0$) i d'esquerra ($j = i+1$) a dreta ($j = 2n-1$), un cop posem a 0 totes les components T_{ij} amb $i \geq j$.

El cost de l'algorisme és $\Theta(n^2)$ tant en temps com en espai, un cop realitzat el preprocés que ens dóna el vector a (amb cost $\Theta(n)$ en temps i en espai) .