

3. (3.5 puntos) La pandemia Covid-19 ha forzado a muchos restaurantes a digitalizarse. El restaurante *Come Sano* dispone de n plazas y conoce las distancias d_{ij} entre cada dos plazas i y j , $0 \leq i, j \leq n - 1$. Tiene reservas para $m \leq n$ comensales y una matriz de booleanos c_{kl} le indica si dos comensales k y l , $0 \leq k, l \leq m - 1$ son o no allegados. Dos personas allegadas pueden sentarse a cualquier distancia, pero dos no allegados han de estar separados al menos dos metros entre sí. Se quiere decidir cómo sentar a todos los comensales en las plazas del restaurante de forma que se respeten las distancias de seguridad y se *maximice* el número de parejas de comensales allegados donde los dos componentes están sentados a menos de dos metros de distancia.

Nota: se entiende que la pareja formada por el comensal k y el comensal l es la misma pareja que la formada por l y k , así que solo se cuenta una vez. Nadie es allegado de sí mismo. Las distancias entre plazas y la relación de ser allegado son simétricas.

- (0,25 puntos) Define el *espacio de soluciones* e indica cómo es el *árbol de exploración*.
- (2.25 puntos) Implementa un algoritmo de *vuelta atrás* que resuelva el problema. Explica claramente los *marcadores* que has utilizado.
- (1 punto) Plantea al menos una *función de poda de optimalidad* e impleméntala en tu algoritmo.

Entrada

La entrada comienza con una línea que contiene el número de casos de prueba. Cada caso de prueba contendrá inicialmente el valor del número de plazas del restaurante n y de comensales m . A continuación n filas con las distancias d_{ij} entre las plazas del restaurante. Y finalmente m filas que indican mediante 0s (falso) y 1s (cierto) si dos comensales son o no allegados.

Salida

Por cada caso de prueba el programa escribirá **CANCELA** si no se puede sentar a los comensales respetando las distancias de seguridad, y en caso contrario escribirá **PAREJAS** seguida de la cantidad máxima de parejas de comensales allegados sentados a menos de dos metros de distancia.

Entrada de ejemplo

```
3
5 4
0 1 1 4 5
1 0 1 3 4
1 1 0 2 3
4 3 2 0 1
5 4 3 1 0
0 1 1 0
1 0 1 0
1 1 0 0
0 0 0 0
2 2
0 0.5
0.5 0
0 0
0 0
4 4
0 0.5 3.5 4
0.5 0 2.5 3
3.5 2.5 0 3
4 3 3 0
0 1 1 0
1 0 1 0
1 1 0 0
0 0 0 0
```

Salida de ejemplo

PAREJAS 3
CANCELA
PAREJAS 1