

11. Un camarero trabaja en un restaurante que dispone de mesas para dos, cuatro y seis comensales. Suponiendo un aforo limitado y que no es posible reservar una mesa con antelación, diseñe un algoritmo voraz que asigne las mesas a los diferentes grupos de comensales que desean acceder al recinto. Suponga que todos desean acceder al mismo tiempo y que el número de grupos y de comensales en cada grupo es desconocido hasta el momento en el que entran en el restaurante y el camarero les tiene que asignar una mesa (los grupos entran de forma ordenada). ¿Es la estrategia devoradora diseñada óptima? Describa los elementos que lo identifican como perteneciente al esquema general de los algoritmos voraces.

Sea C: conjunto de grupos de comensales y K: conjunto de mesas

Restaurante: $C \times K \rightarrow S$

inicio

$S \leftarrow \emptyset$

$M \leftarrow K$

Mientras $C \neq \emptyset \wedge M \neq \emptyset$

$g \leftarrow \text{siguiente_comensal}(C)$

$C \leftarrow C - \{g\}$

mientras $g > 0 \wedge M \neq \emptyset$

$\text{mesa} \leftarrow \text{seleccionar_mesa}(g, M)$

$g \leftarrow g - \text{mesa}$

$S \leftarrow S \cup \{\text{mesa}\}$

$M \leftarrow M - \{\text{mesa}\}$

fin

Seleccionar_mesa: $g \times M \rightarrow \text{valor}$

$\text{valor} = 0$

si $g \leq 6$

para todo $a \in \{\text{mesas}\}$

si $a \geq \text{comensales} \wedge \text{valor} < a$

$\text{valor} \leftarrow a$

si $\text{valor} = 0$

$\text{valor} \leftarrow \max(\text{mesa})$

- 12.** Resuelva el ejercicio anterior suponiendo que se conoce previamente el tamaño de los grupos de comensales interesados en acceder al restaurante. ¿Es la estrategia devoradora diseñada óptima?

Sea C: conjunto de grupos de comensales y K: conjunto de mesas

Restaurante: Comensales x Mesas -> S

inicio

S <- \emptyset

M <- Mesas

C <- Comensales

Ordenar(C)

Mientras C $\neq \emptyset$ ^ M $\neq \emptyset$

 g <- siguiente_comensal (C)

 C <- C - {g}

 mientras g > 0 ^ M $\neq \emptyset$

 mesa <- seleccionar_mesa(g,M)

 g <- g - mesa

 S <- S \cup {mesa}

 M <- M - {mesa}

fin