# Auxiliar 6 - Algoritmos Online

CC4102 - Diseño y Análisis de Algoritmos
Profesor: Gonzalo Navarro
Auxiliar: Jorge Bahamonde
Ayudantes : Sebastián Ferrada, Willy Maikowski

1. Considere un escenario donde tiene $k$ *servidores* en un *espacio métrico* (donde está definida una función de distancia $d$ simétrica, no negativa y que cumple la desigualdad triangular) y una secuencia de *peticiones* (puntos en este espacio) que debe atender. Cada vez que llega una petición, un servidor debe moverse hacia esa posición; el costo de una solución consiste en la distancia total recorrida por todos los servidores.

   Un algoritmo *online* para este problema no conoce la secuencia completa de puntos a atender, mientras que el *offline* sí.

   (a) Muestre que para el caso de 2 servidores en una línea, la idea "obvia" de mover siempre el servidor más cercano resulta en una competitividad no acotada.

   (b) Considere el siguiente algoritmo:

   - Si ambos servidores están al mismo lado de la petición, se envía el servidor más cercano.
   - Si la petición está entre los dos servidores, se envían los dos a velocidad constante, deteniéndose cuando uno de ellos llega al objetivo.

   Muestre que este algoritmo es 2-competitivo.

   (c) Sea $\mathcal{A}$ un algoritmo online para el problema de los $k$ servidores bajo un espacio métrico arbitrario con al menos $k+1$ puntos. Pruebe que $\mathcal{A}$ es al menos $k$-competitivo.

   (d) **Propuesto:** Extienda el algoritmo de la parte (b) a $k$ servidores en una línea.

2. Tenemos $m$ procesadores idénticos. Como entrada recibimos una lista $t_1$, $t_2$,...,$t_n$ de tareas con tiempos de procesamiento $p_1$, $p_2$,...,$p_n > 0$, respectivamente. Las tareas son recibidas secuencialmente, y sólo cuando una tarea llega conocemos su tiempo de procesamiento. Cada tarea debe ser asignada a una máquina inmediatamente, y la decisión no puede ser cambiada. La *carga* de un procesador es la suma de los tiempos de procesamiento de todas las tareas que le son asignadas. El costo de un algoritmo que resuelve el problema de asignacion es la máxima carga entre sus procesadores.

   Considere el siguiente algoritmo para el problema anterior. Cada tarea es asignada al procesador con menor carga (en caso de empate, se elige cualquiera).

   (a) Demuestre que este algoritmo es $(2 - \frac{1}{m})$-competitivo.

   (b) Demuestre que esta cota es óptima para el algoritmo.