Algoritmo A: elijo a los atletas i

en base a su $d_i = b_i + c_i$ en forma decreciente.

$n_i \to$ tiempo que tarda en nadar el atleta i

$b_i \to$ tiempo que tarda en andar en bici el atleta i

$c_i \to$ tiempo que tarda en correr el atleta i

El primer ejercicio lo probamos con el lema "Stays ahead"

este lo haremos con el exchange argument.

Luego suponemos que existe un Alg O óptimo distinto de A. Entonces necesariamente habrá al menos un caso donde i se ligue antes que j t.q $d_i \leq d_j$. A (i,j) le llamaremos inversión.

La idea es demostrar que podemos revertir esas inversiones sin afectar el tiempo de la competencia y sin generar nuevas inversiones. demostrando que A (nuestro algoritmo sin inversiones) es óptimo.

Primero, necesitamos demostrar que al existir dicha inversión entre i y j, necesariamente existe una inversión entre k_1 y k_2 consecutivos t.q $i \leq k_1 \leq j$

$i \leq k_2 \leq j$

$k_1 \neq k_2$

Si recorremos desde i hasta j, tenemos dos opciones con respecto a los d de los atletas. Vemos que los d disminuyen/son iguales hasta que al pasar a cierto atleta su d aumenta (generando una inversión consecutiva) o, en el otro caso, siguen disminuyendo/manteniendose. Sin embargo, en este segundo caso, al llegar a j (como $d_j > d_i$) necesariamente habrá una inversión entre j y $j-1$ (consecutivos) ■

Habiendo demostrado esto, resta ver que eliminando estas inversiones no se generan nuevas y el tiempo de la competencia no se ve afectado.

Veamos primero que el tiempo de la competencia no crece

Previo a deshacer la inversión tenemos que $T_{pre} = n_i + n_j + b_j + c_j$ (no se toma en cuenta $b_i + c_i$ porque $b_i + c_i \leq b_j + c_j$ y como además i empieza antes entonces termina antes que j)

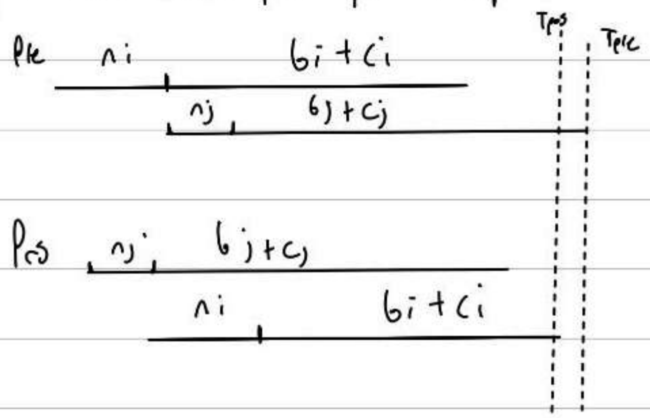
(T = tiempo)

Pero luego de deshacerlo tenemos $T_{pos} = n_j + n_i + \max\{b_i + c_i, b_j + c_j - n_i\}$

Como $b_i + c_i \leq b_j + c_j$, en ambos casos $T_{pos} \leq T_{pre}$.

Por lo tanto concluimos que no empeora el tiempo

Ejemplo:



el tiempo de otros atletas $k / k < i$ no se modifica ya que en ambos casos se comienza en el mismo momento (Pre o Pos) y para los $k / k > j$ sucede lo mismo o mejor

(en el caso $T_{pos} < T_{pre}$ los $k / k > j$ pueden empezar antes y por ende terminar antes)

Por lo tanto este procedimiento no afecta la optimalidad del algoritmo (no empeora el tiempo de la competencia) ■

Ahora resta ver que no generamos otras inversiones.

Supongamos sin pérdida de generalidad que había una única inversión,

luego para todo $k / k > j$ se cumple $d_k < d_j$ $d_k < d_i$

y para todo $k' / k' < i$ se cumple $d_{k'} > d_j$ $d_{k'} > d_i$

Como el único swap fue entre i y j y fue local (entre ambos siendo consecutivos)

estas desigualdades se siguen cumpliendo y por lo tanto se vislumbra que no hay nuevas inversiones generadas ■

Luego repitiendo el procedimiento de deshacer todas las inversiones obtenemos el mismo resultado que nuestro Algoritmo A (sin inversiones) probando así que este es óptimo ■