

# Comparación Problema de Selección

Nicolás Camacho Plazas

6 de octubre de 2020

## Complejidad Espacial

- Versión Programación Dinámica: Una de las características esenciales de los algoritmos de programación dinámica es que, mediante el uso exhaustivo de la memoria (tabla de memoización y backtracking), se busca reducir el tiempo de procesamiento. La versión de programación dinámica, por su matriz de memoización y la de backtracking, tiene una complejidad espacial  $O(n^2)$ , siendo  $n$  la cantidad de actividades.
- Versión “Voraz”: Puesto que, en los algoritmos voraces, al tomar decisiones tacañas, se reducen los subproblemas a un único problema, no es necesaria una matriz, de modo que la complejidad espacial es  $O(n)$ , siendo  $n$  la cantidad de actividades.

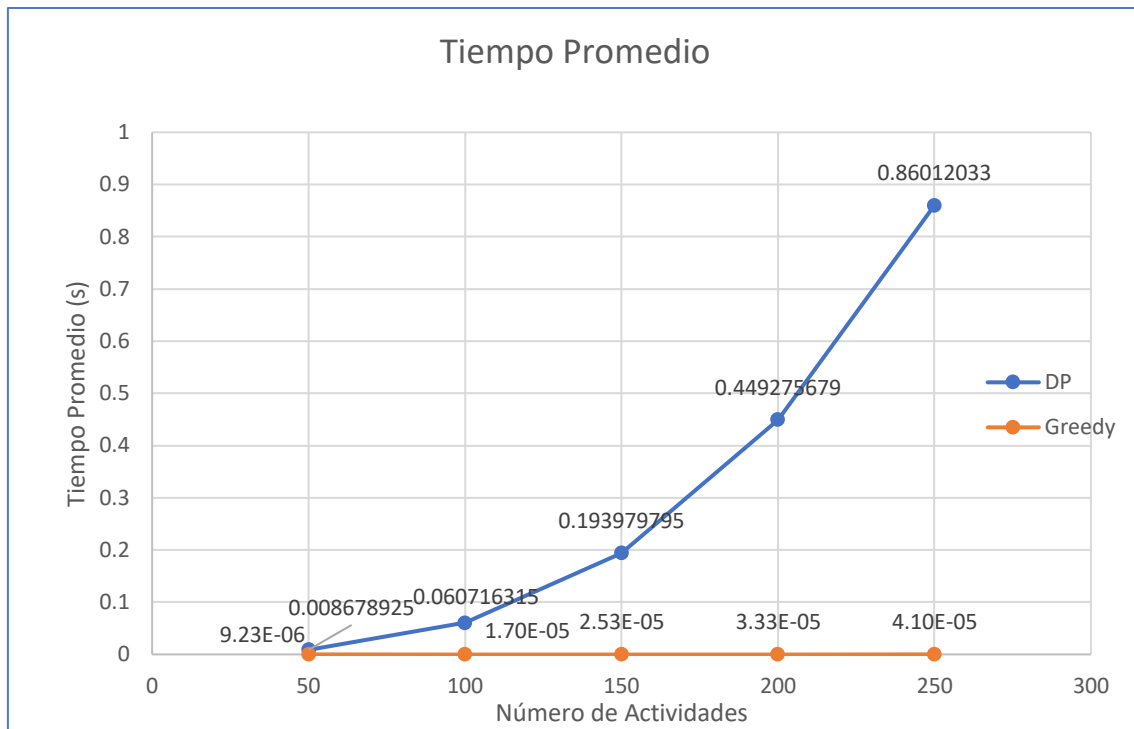
## Complejidad Espacial

- Versión Programación Dinámica: Tras realizar inspección de código se concluyó que el algoritmo tiene un orden de complejidad temporal de  $O(n^3)$ , siendo  $n$  la cantidad de actividades.
- Versión Programación Dinámica: Tras realizar inspección de código se concluyó que el algoritmo tiene un orden de complejidad temporal de  $O(n)$ , siendo  $n$  la cantidad de actividades.

## Tiempo promedio

Para comparar la velocidad de cada versión, se configuraron cinco escenarios de prueba (50 actividades, 100 actividades, 150 actividades, 200 actividades, 250 actividades). En cada intento se generaba una lista de actividades aleatorias cuyo inicio y fin podían transcurrir en un rango de 24 horas; los algoritmos se ejecutaron con las mismas actividades y se midió su tiempo de ejecución. Se guardó el tiempo promedio en 20 ejecuciones en cada escenario de prueba obteniendo los siguientes resultados:

Tabla 1 - Tiempo promedio (20 ejecuciones por número de actividades)



Se puede apreciar la enorme diferencia en donde en cada intento, y por un margen gigantesco, el algoritmo Voraz presentó un desempeño superior. Estos resultados son consistentes con el orden de complejidad calculado previamente.

### Calidad de la solución

Sorprendentemente, y pese a que la teoría establece que los algoritmos voraces sacrifican calidad de la solución por tiempo de ejecución y espacio ahorrado, en las 100 ejecuciones (20 por cada tamaño mencionado previamente), los dos algoritmos arrojaron resultados con la misma calidad, es decir, llegaban al mismo número de actividades cada vez.

```
----- Selector Solution Test -----  
For 20 runs with 50 activities...  
Times when DP offers a better solution: 0  
Times when Greedy offers a better solution: 0  
Ties: 20  
For 20 runs with 100 activities...  
Times when DP offers a better solution: 0  
Times when Greedy offers a better solution: 0  
Ties: 20  
For 20 runs with 150 activities...  
Times when DP offers a better solution: 0  
Times when Greedy offers a better solution: 0  
Ties: 20  
For 20 runs with 200 activities...  
Times when DP offers a better solution: 0  
Times when Greedy offers a better solution: 0  
Ties: 20  
For 20 runs with 250 activities...  
Times that DP give a better solution: 0  
Times when Greedy offers a better solution: 0  
Ties: 20
```