

Programación 3

Ejercicio entregable del TP 4

Contexto

Elon Musk decide abrir su taller espacial para permitirle a las familias que puedan ir a conocerlo. Luego de abierta la inscripción, 5000 familias se anotan para realizar la visita. Debido a la gran demanda de público decide establecer un número de 100 días durante los cuales aceptará visitas. Sabiendo que el taller tiene una capacidad diaria acotada (de **340** personas), Elon debe distribuir a los visitantes durante los 100 días. Para esto, solicita a cada familia que le informe **8** días (en orden de preferencia) en los cuales preferirían realizar la visita, y cuántas personas conforman el grupo familiar.

Por ejemplo, la familia Pérez tiene 3 miembros, y sus preferencias son [33, 25, 10, 62, 48, 53, 66, 78]. El día 33 es el día preferido para los Perez, de no ser posible ir el 33, el segundo día preferido es el 25, y así sucesivamente.

Dado que no todas las personas podrán asistir el día que prefieren, Elon decide crear un bono compensatorio para las familias que no son asignadas al día que eligieron como prioritario.

Elon nos pide que diseñemos un programa que permita determinar qué familias deberán ser invitadas en cada uno de los 100 días para **minimizar el monto total** de los bonos compensatorios, considerando que: todas las familias deben poder visitar el taller en alguno de los 8 días que indicaron como preferidos, y cada familia será invitada un único día.

Datos de entrada

Se anotaron 5000 familias para visitar el taller, cada una eligió entre 8 días de los disponibles en el orden en que prefieren realizar la visita.

Se sabe que el taller estará abierto 100 días consecutivos y que la disponibilidad diaria es de 340 personas.

El bono compensatorio se aplica únicamente a la familia que no es asignada en el primer día elegido como prioritario y se calcula utilizando la siguiente fórmula:

$$\text{bono} = \text{U\$S } 25 + (\text{U\$S } 10 * \text{Grupo Familiar}) + (\text{U\$S } 5 * \text{Dia asignado})$$

Características de la solución

Se considera válida a una solución si:

- No supera el tope de capacidad diaria para ninguno de los días disponibles.
- Cada familia es asignada a un único día para realizar la visita.
- Todas las familias son asignadas para realizar la visita en algunos de sus días preferidos.

Se considera óptima a una solución si:

- Es una solución válida.
- Se consigue obtener el mínimo costo total del bono compensatorio.

Se sabe que, para el caso de prueba provisto, el mínimo costo total del bono compensatorio es **29035**.

Consigna

1. Implementar un programa que obtenga una solución válida al problema planteado, resultando en el cronograma de familias por día y calcule el monto total a disponer en bonos compensatorios.
2. Realizar un proceso que, a partir de una solución válida, intente mejorar esa solución mediante alguna transformación local cuyo costo computacional no supere una función cuadrática - $O(N^2)$.
3. Realizar un breve informe que contenga:
 1. Las ideas que se consideraron para obtener una solución válida (incluyendo aquellas que fueron descartando en el proceso).
 2. Una explicación del algoritmo que se utilizó para obtener una solución válida, justificando su elección.
 3. Las transformaciones aplicadas a la solución válida, justificando su elección.

Las ideas que se consideraron para obtener una solución válida (incluyendo aquellas que fueron descartando en el proceso)

La primera idea que se me ocurrió para obtener una mejor solución fue la de tener un orden en el arreglo familias según su cantidad de miembros de manera descendente. Y fue porque creía que al meter a las familias de 8 miembros para abajo el costo iba a ser menor y el resultado fue que:

Cuando la cantidad de miembros era de mayor a menor el bono mínimo que me calculaba era de \$48.135.

Cuando buscaba crear el cronograma sin ningún orden el bono mínimo quedaría de \$40.845.

Entonces pensé que eso se debía a que al darle tanta prioridad a las familias de 8 miembros estaba llenando muy rápido el cupo de 340 personas en los días preferidos, entonces cambie el método de ordenamiento para que ordene el arreglo familia según sus miembros de manera ascendente, entonces el costo se redujo hasta \$36.660.

Después quise ver si a ese resultado podría modificarlo para obtener un mejor resultado, entonces modifique el criterio para que cuando dos familias posean la misma cantidad de miembros los ordene según su día preferido de forma ascendente. Logrando así un bono de \$35.430.

Una explicación del algoritmo que se utilizó para obtener una solución válida, justificando su elección

Una vez que las familias estaban ordenadas con el criterio mencionado anteriormente, use un algoritmo en el que se va recorriendo el arreglo familia por familia y en cada una pregunta si esta puede ser asignado en su día favorito, es decir, si para ese día todavía queda un cupo para la cantidad de sus miembros, de no ser así pregunta para su segundo día y así sucesivamente.

Elegí solo recorrer el arreglo de familias porque suponía que el ordenamiento del criterio cumpliría con las expectativas de una buena solución y, en el caso de querer mejorar los resultados solo debería crear un nuevo criterio de ordenamiento y el algoritmo seguiría funcionando igual. Es decir, la selección depende en gran medida por cómo viniera ordenado el arreglo.

Las transformaciones aplicadas a la solución válida, justificando su elección

La transformación local la elegí porque puede darse el caso que en un día X tenga todavía lugares disponibles, entonces es más rentable tomar una familia de ese día con pocos miembros y agarrar otra de los días siguientes con más miembros y si esta no se pasa del cupo máximo del día X y es su día preferido entonces hace el cambio.

Por ejemplo, en el día 34 se asignaron 336 personas, entonces podría darse el caso que sacando una familia de 2 personas y agregando otra de 6 miembros tenga mejor resultado. De esa manera llega a un bono de \$35.160, es decir, \$270 menos.