Ejercicio 1. Demostrar que si se incluye una operación de *decrementar una unidad* en el contador binario visto en clase en el tema introductorio de análisis amortizado, n operaciones cuestan como mucho $\Theta(nk)$ en tiempo.

Ejercicio 2. Se realiza una secuencia de n operaciones sobre una estructura de datos dada. La operación i-ésima cuesta i si i es una potencia exacta de 2, y 1 en otro caso. Calcular mediante el método agregado el coste amortizado de cada operación.

Ejercicio 3. Considerar un montículo (ordinario) con n elementos y con las operaciones de inserción y borrado del mínimo en $O(\log n)$ en el caso peor. Dar una función de potencial P tal que el coste amortizado de insertar sea $O(\log n)$ y el coste amortizado de borrar el mínimo sea $O(1)$, y ver que es válida como función de potencial.

Ejercicio 4. Describir cómo implementar una cola usando dos pilas de manera que el coste amortizado de cada operación de insertar en la cola y de borrar de la cola sea $O(1)$.

Ejercicio 5. Suponer que en lugar de contraer una tabla *hash* dividiendo por dos su capacidad cuando su factor de carga queda por debajo de 1/4, la contraemos multiplicando por 2/3 su capacidad cuando su factor de carga queda por debajo de 1/3. Usando la función de potencial $P(T) = |2 \cdot \mathit{numdatos}(T) - \mathit{capacidad}(T)|$, demostrar que el coste amortizado de la operación de borrado usando esa estrategia de contracción está acotado por una constante.

Ejercicio 6. En la implementación de conjuntos disjuntos con listas encadenadas de las transparencias 50-51, hace falta un puntero al primer elemento y otro al último para poder implementar la unión con un coste lineal en la longitud de la segunda lista. El profesor Metodio sospecha que podría ser posible almacenar un único puntero en lugar de dos (manteniendo dos punteros en cada registro de la lista). Demostrar que la sospecha del prof. Metodio puede estar bien fundada describiendo cómo representar cada conjunto con una lista encadenada y de forma que cada operación (*crear*, *encontrar*, *unir*) tenga el mismo coste que los descritos en las transparencias 50-51. Describir también cómo serían esas operaciones. El esquema debe permitir implementar la heurística de la unión de la transparencia 53, con el mismo efecto que el descrito allí. (Idea: usar la cola de una lista encadenada como representante del conjunto.)

Ejercicio 7. Implementar una versión no recursiva de la operación *encontrar* para una estructura de conjuntos disjuntos usando un conjunto de árboles con la heurística de compresión de caminos.

Ejercicio 8. Suponer que queremos añadir a una estructura de conjuntos disjuntos una operación de *escribirConjunto*(x) que, dado un nodo x, escribe todos los miembros del conjunto de x, en cualquier orden. Mostrar cómo podemos añadir un único atributo a cada nodo del bosque de conjuntos disjuntos de forma que *escribirConjunto*(x) tenga un coste lineal en el número de miembros del conjunto de x, y el coste asintótico de las otras operaciones no cambie. Suponer que la escritura de cada elemento del conjunto tiene coste $O(1)$.

Si entregas alguno de estos ejercicios, debes hacerlo no más tarde del 20-12-2013.