

Reflexión Act 1.3 Juan Pablo Salazar García – A01740200

Este problema fue bastante sencillo, ya que al separarlo en partes se vuelve una serie de problemas sencillos con soluciones fáciles de implementar.

Para la parte del sort elegimos utilizar el sort que viene incluido en c++ ya que es rápido ($O(n \log n)$), y tiene funcionalidades que nos facilitaron muchísimo el acomodar la lista ya que puedes usar un parámetro para que el sort use para hacer el sort, que en este caso era la fecha en mes y día sumados.

En la parte de la búsqueda, ya que teníamos una lista con índices definidos (fecha), teníamos un caso perfecto para usar búsqueda binaria. Preferimos usar este tipo de búsqueda ya que es muy rápida siendo $O(\log n)$. Primero buscamos el primer valor de la primera fecha que se nos diera, y después buscamos el primero de la fecha que sigue de la segunda fecha que se nos diera, y se le restaba uno para conseguir el último de la que se nos pidió.

El uso de los algoritmos de búsqueda y ordenamiento en esta actividad fue simple, pero el que entendiéramos y supiéramos como funcionan estos nos guiaron a elegir los algoritmos que elegimos, que tienen una eficiencia extremadamente alta en comparación con los básicos como bubble sort o búsqueda lineal, que son los que uno usaría si no conociera estos algoritmos bien lo suficiente como para saber que hay alternativas muchísimo mejores en todos los aspectos.