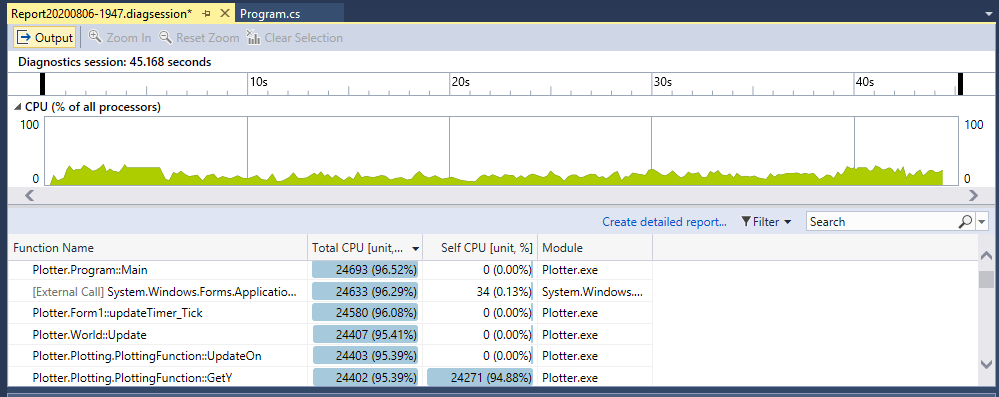
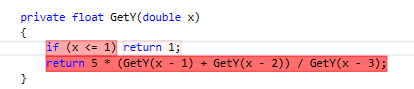
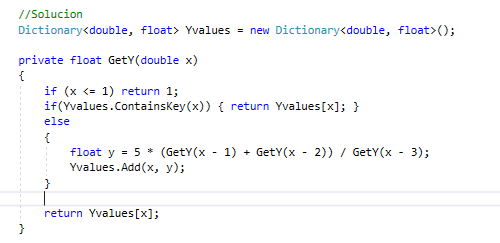
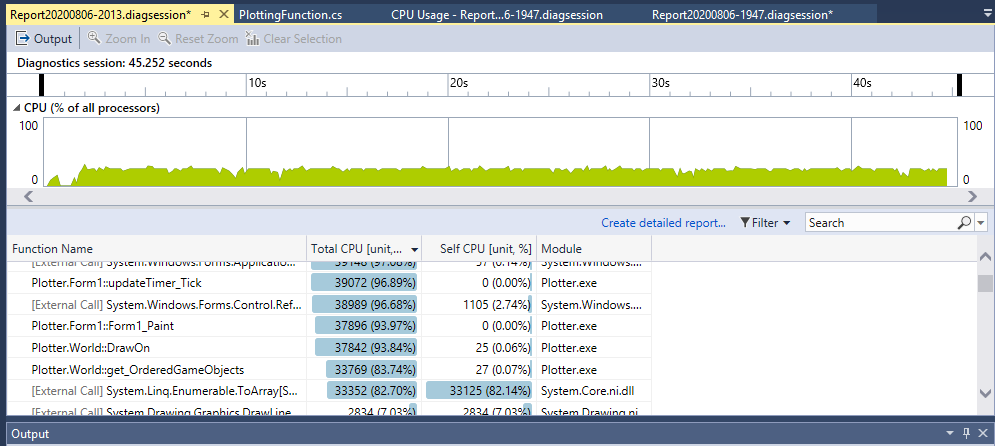
1. La aplicación grafica muy lentamente los primeros intervalos de la función F(n) y se ralentiza de forma tal que parece se hubiera colgado.
2. Habiendo corrido un Profiler de CPU se puede determinar que el problema principal a atacar esta localizado en la función GetY(x)  
   
3. El problema se debe a que esta función se resuelve de forma recursiva, algo que esta comprobado como poco performante. Para poder hacer GetY(x-2), debe hacer GetY(x-1) y lo mismo para (x-3), pasara de nuevo por (x -2) y (x-1). Lo cual hace una cantidad de llamadas absurdamente grande a esta función.  
   
4. A priori se podría ver 2 posibles soluciones, la primera seria reescribir esta instrucción de manera interativa, lo cual sin lugar a duda aliviaría el call stack.

La segunda solución viable seria guardar en una colección del tipo Dicitionary los resultados de los X anteriores, permitiendo cachear de esta forma la operatoria. Si bien se utiliza memoria, se gana un montón en velocidad de proceso, dado que el big O de acceso y escritura de una Hashtable es del orden de magnitud de O(1). Ref: biocheatsheet.com

1. Decido implementar la segunda opción, ya que es más rápida de implementar.  
   Refactoreo la función de la siguiente forma.  
     
     
   Corriendo un segundo profiler queda demostrado que la función ya no es un hotspot del programa.  
     
     
   Habiendo que scrollear para volver a encontrarla con un uso despreciable.  
   