Programación II. Curso 2019-2020
Trabajo Obligatorio

Introducción

En este trabajo obligatorio se propone especificar y diseñar algoritmos para resolver el problema del máximo solape entre intervalos. Dado un conjunto de intervalos definidos en los reales, calcular el máximo solape consiste en encontrar la pareja de intervalos cuyo solape es máximo. Por ejemplo, en el conjunto de intervalos $\{[1.5, 8.0], [0.0, 4.5], [2.0, 4.0], [1.0, 6.0], [3.5, 7.0]\}$, que se representan gráficamente en la Figura 1, el máximo solape es el de los intervalos $[1.5, 8.0]$ y $[1.0, 6.0]$ que es igual a 4.5.

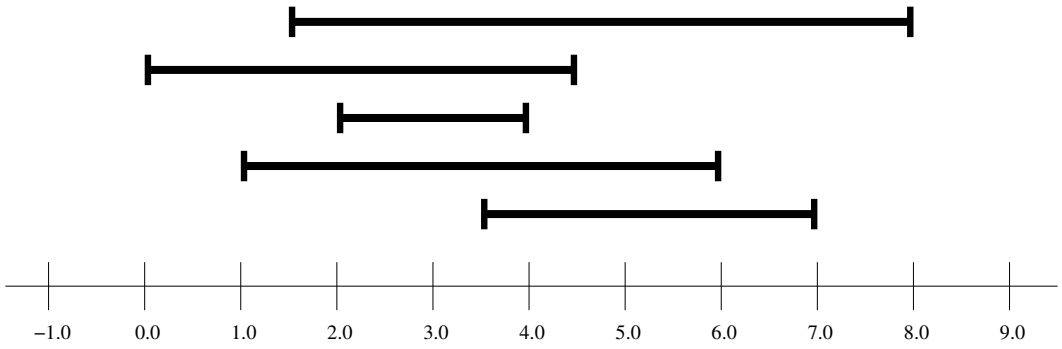

Figura 1: Intervalos definidos en los reales.

Tareas

Se propone resolver el problema del máximo solape mediante dos estrategias distintas: *Fuerza Bruta* y *Divide y Vencerás*.

Se pide realizar las siguientes tareas:

- Especificar formalmente las funciones declaradas en el fichero `maxsolape.hpp`.
- Diseñar e implementar la estrategia de *Fuerza Bruta* para resolver el problema. El coste asintótico en tiempo será $\mathcal{O}(n^2)$ donde n es el número de intervalos.
- Diseñar e implementar la estrategia de *Divide y Vencerás* para resolver el problema. El coste asintótico en tiempo será $\mathcal{O}(n \cdot log(n))$ donde n es el número de intervalos.

- Ejecutar ambas estrategias para conjuntos de entre 100 y 4000 intervalos generados de forma aleatoria (el valor mínimo para el extremo izquierdo del intervalo es `minini` y el valor máximo para el extremo derecho `maxfin`, ver `maxsolape.hpp`).
- Generar gráficas que muestren el tiempo de ejecución en función del número de intervalos para cada una de las estrategias. Las gráficas se pueden realizar de la manera que se desee. Una posibilidad consiste en guardar las datos que se quieren mostrar en un fichero de texto y después realizar una llamada al sistema desde el programa. Por ejemplo, si se han guardado los siguientes datos en el fichero de texto `tfb.txt` (cada línea representa el número de intervalos y el tiempo de ejecución en microsegundos):

  ```
  100     2260
  150     5180
  200     9520
  250     16280
  300     14580
  ```

  entonces la instrucción

  ```
  system("gnuplot -e \"set terminal gif; set style data lines; plot 'tfb.txt'\" > tfb.gif");
  ```

  realiza una llamada al sistema que ejecuta `gnuplot` para generar una gráfica con los datos de `tfb.txt` en el fichero `tfb.gif`. Puedes encontrar información sobre *gnuplot* en `http://www.gnuplot.info/`
- Describir brevemente en un fichero `comentarios.pdf` las gráficas obtenidas y una concisa discusión sobre las ventajas de cada estrategia.

Notas:

- En la página web de la asignatura se proporcionan los ficheros `maxsolape.hpp` y `Makefile` que asume que el programa principal está implementado en el fichero `costemsolap.cpp`.
- Aunque se puede hacer el uso que se desee de estos ficheros, obligatoriamente deberán implementerase la funciones especificadas en el fichero `maxsolape.hpp` original.

Forma y fecha de entrega

- Los apartados relativos al diseño se entregarán vía *moodle*.
- Se deberá entregar un fichero con nombre `AAAAAA.zip`, donde AAAAAA es tu NIA, que contenga los siguientes ficheros:
  - Todos los ficheros `.hpp` y `.cpp`.
  - El fichero `Makefile` que los compila y genera el ejecutable `maxsolape`.
  - El fichero `comentarios.pdf` que describe brevemente los costes temporales de las estrategias.

- Las especificaciones formales deben escribirse en el fichero `maxsolape.hpp`.
- El último día para entregar el trabajo es el viernes 5 de junio de 2020.

Sobre el presente trabajo obligatorio

La evaluación de la asignatura Programación 2 en la primera convocatoria consta de tres pruebas:

- Examen escrito con un peso del 70 %. Para aprobar la asignatura es necesaria una calificación mínima de 4.0 puntos en él.
- Examen práctico en laboratorio con un peso del 15 %. No se exige calificación mínima.
- El presente trabajo obligatorio con un peso del 15 %. No se exige calificación mínima.