Pràctica 1: Algorismes i eficiència (10–14 de març, 2003)

Práctica 1: Algoritmos y eficiencia (10–14 de marzo, 2003)

Objectiu: Estudi comparatiu d'algorismes senzills i caracterització dels seus costs temporals mesurats en passos.
Objetivo: *Estudio comparativo de algoritmos sencillos y caracterización de sus costes temporales medidos en pasos.*

Es consideraran en aquesta pràctica 3 algorismes de cerca en vectors ordenats: cerca lineal, cerca dicotòmica i cerca dicotòmica *modificada*. Caldrà implementar els tres algorismes com a funcions que prenen un valor x, un vector v i una talla n i tornen un índex entre 0 i n (n significa que l'element x no es troba en v).

Se considerarán en esta práctica 3 algoritmos de búsqueda en vectores ordenados: búsqueda lineal, dicotómica y dicotómica modificada*. Habrá que implementar los tres algoritmos como funciones que toman un valor x, un vector v y una talla n y devuelven un índice entre 0 y n (n significa que el elemento x no se encuentra en v).*

Caldrà fer un programa principal que genere un vector ordenat i al que se li puga donar un valor x. El programa haurà de deixar clar que els algorismes implementats funcionen correctament.

Habrá que hacer un programa principal que genere un vector ordenado y al que se pueda dar un valor x. El programa tendrá que dejar claro que los algoritmos implementados funcionan correctamente.

L'algorisme de cerca lineal no mereix cap comentari. La versió de la cerca dicotòmica que s'ha estudiat en teoria, és:

El algoritmo de búsqueda lineal no merece ningún comentario. La versión de la búsqueda dicotómica que se ha estudiado en teoría es:

```
int BusBinClas (Vector v, int n, int x)
// cerca x en el vector v de n elements.
// Torna l'índex on es troba x o n si no està
{
   int izq, der, cen, aux;

   izq = 0;
   der = n - 1;
   cen = (izq + der) / 2;

   while ( (izq < der) && (v[cen] != x) )
   {
      if (v[cen] > x)
         der = cen - 1;
      else
         izq = cen + 1;
      cen = (izq + der) / 2;
   }

   if (v[cen] == x)
      aux = cen;
   else
      aux = n;

   return aux;
}
```

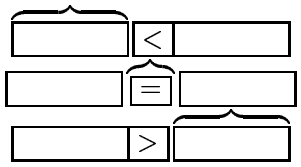


Figura 1

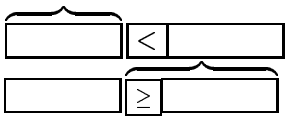


Figura 2

L'algorisme anterior rep el nom de cerca dicotòmica de tres vies ja que en cada pas es poden prendre tres decisions: esquerra, dreta o acabar (Figura 1). Una alternativa és l'anomenada cerca dicotòmica de dues vies (*modificada*). L'algorisme consisteix a continuar la cerca per l'esquerra o per la dreta incloent en una de les dues opcions també la posició central independentment de si conté o no l'element x (Figura 2). Encara que aquest algorisme no té un millor cas com l'anterior, és evident que requereix una comparació **menys** en **cada** iteració. (Atenció en la possibilitat d'entrar en un bucle infinit en aquest algorisme!)

*El algoritmo anterior recibe el nombre de búsqueda dicotómica de tres vias ya que en cada paso se pueden tomar tres decisiones: izquierda, derecha o acabar (Figura 1). Una alternativa es la llamada búsqueda dicotómica de dos vias (*modificada*). El algoritmo consiste en continuar la búsqueda por la izquierda o por la derecha incluyendo en una de las dos opciones también la posición central independientemente de si contiene o no el elemento x (Figura 2). Aunque este algoritmo no tiene un mejor caso como el anterior, es evidente que requiere una comparacion* **menos** *en* **cada** *iteración. (Cuidado con la posibilidad de entrar en un bucle infinito con este algoritmo!)*

Estudi comparatiu / *Estudio comparativo*

Es tracta d'avaluar els costs dels tres algorismes anteriors. La instrucció crítica que considerarem serà la comparació sobre elements del vector (no les comparacions entre índexs). S'hauran d'afegir comptadors de comparacions dins dels algorismes. Per tal de calcular el cost mitjà, caldrà repetir cerques aleatòries (`rand()`) sobre un mateix vector i calcular el promig.
Per a cada talla, el vector serà fix i contindrà enters senars (1,3,5,...). Aleshores considerarem dos casos, a) generació i cerca d'enters aleatoris entre 0 i $2n$ (50 % de probabilitat d'èxit) i b) generació aleatòria d'índexs i entre 0 i $n-1$ i cerca del valor $x = v[i]$ (100 % de probabilitat d'èxit).

Se trata de evaluar los costes de los tres algoritmos anteriores. La instrucción crítica considerada será la comparación sobre elementos del vector (no las comparaciones de índices). Habrá que añadir contadores de comparaciones dentro de los algoritmos. Para calcular el coste medio, habrá que repetir búsquedas aleatorias (`rand()`*) sobre un mismo vector y calcular el promedio.*
Para cada talla, el vector será fijo y contendrá enteros impares (1,3,5,...). Consideraremos dos casos a) generación y búsqueda de enteros aleatorios entre 0 *y* $2n$ *(50 % de probabilidad de éxito) i b) generación aleatoria de índices* i *entre* 0 *i* $n-1$ *y búsqueda del valor* $x = v[i]$ *(100 % de probabilidad de éxito).*

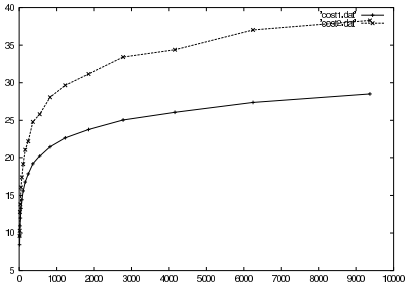
Es podrà considerar un únic vector de la talla màxima i considerar la part corresponent per a cada n (compte amb els índexs fóra de rang). Per tal d'unificar les diferents talles a considerar es farà servir un bucle de la forma `for (n = 10; n <10000; n = n*3/2 )` i per cada talla, el nombre de cerques haurà de ser d'almenys 5 vegades n.

Se podrá considerar un único vector de la talla máxima y considerar la parte correspondiente para cada n *(cuidado con los índices fuera de rango). Para unificar las diferentes tallas a considerar habrá que usar un bucle de la forma* `for (n = 10; n <10000; n = n*3/2 )`*. Para cada talla, el número de búsquedas deberá ser de al menos 5 veces* n*.*

Si el programa escriu en un fitxer els resultats en dues columnes (per a n i per a $T(n)$, respectivament) aleshores es pot mostrar la funció cost gràficament mijantçant el programa `gnuplot`. Alguns comandaments d'aquest són:

Si el programa escribe en un fichero los resultados en dos columnas (para n *y para* $T(n)$*, respectivamente) entonces se puede mostrar la función coste gràficamente mediante el programa* `gnuplot`*. Algunos comandos de éste son:*

```
gnuplot>plot 'cost1.dat' w lp, 'cost2.dat' w lp # mostra dades de 2 fitxers
gnuplot>set data style lp
gnuplot>plot 'cost1.dat', 'cost2.dat'       # idem
gnuplot>save 'graf.plt'        # guarda una gràfica
gnuplot>load 'graf.plt'        # recupera una gràfica
gnuplot>set log y       # eix y en escala logarítmica
gnuplot>set yrange [0:max]        # fixa el rang de l'eix y
```



Documentació a entregar / *Documentación a entregar*

1) fitxers de programes (`pro.cpp, cost.cpp`) desenvolupats corresponents a la comprovació i a l'estudi comparatiu. Cal adjuntar informació suficient per compilar, executar i comprovar que els programes funcionen be.

1) ficheros de programas (`pro.cpp, cost.cpp`*) desarrollados correspondientes a la comprobación y al estudio comparativo. Hay que adjuntar información suficiente para compilar, ejecutar y comprobar que los programas funcionan bien.*

2) funcions cost obtingudes en fitxers separats en el format anteriorment descrit. El nom haurà de ser `ALG-P.dat`, on ALG pot ser `lin`, `bin` o `bim` i P pot ser 50 ó 100 (probabilitat d'èxit).

2) funciones coste obtenidas en ficheros separados con el formato anteriormente descrito. El nombre será `ALG-P.dat`*, donde ALG puede ser* `lin`*,* `bin` *o* `bim` *y P puede ser 50 ó 100 (probabilidad de éxito).*

3) fitxers de gràfiques de gnuplot on es comparen els diferents algorismes. Concretament `binlin-P.plt`, `bin23-P.plt` on P és la probabilitat d'èxit.

3) ficheros de gráficas de gnuplot donde se comparen los diferentes algoritmos. Concretamente `binlin-P.plt`*,* `bin23-P.plt` *donde P es la probabilidad de éxito.*

4) fitxer de tipus text en el que hauràs de comentar els resultats que has obtingut en l'estudi comparatiu. Comenta si et pareixen o no lògics, si eren esperables, etc. Què creus que passarà amb el cost "real" mesurat com a temps d'execució?

4) fichero de tipo texto en el que tendrás que comentar los resultados obtenidos en el estudio comparativo. Comenta si te parecen o lo lógicos, si eran esperables, etc. ¿Qué crees que pasará con el coste "real" medido como tiempo de ejecución?