ESCOLA TÈCNICA SUPERIOR D'ENGINYERIA
Universitat Rovira i Virgili

[DEIM]

L3. El cost d'ordenar

Fonaments de Programació II

Objectius de la pràctica

Els objectius de la pràctica són els següents:

- Revisar l'algorisme d'ordenació per selecció
- Valorar, a nivell empíric, el cost de generar i ordenar una taula d'enters.

La pràctica es fa de forma individual.

Ordenació per selecció

L'ordenació per selecció és un dels mètodes més senzills que tenim per ordenar una taula de dades. En concret farem servir aquest mètode per ordenar una taula d'enters d'una dimensió en **ordre creixent**.

L'algorisme consta de dos bucles: un **bucle extern** que itera per a cadascuna de les posicions a ordenar i un **bucle intern** per trobar l'element menor en les posicions encara no ordenades.

Un cop trobat aquest element, s'intercanvia aquest element amb el de la posició que marca el bucle extern.

```
ordena (n, taula)
  enter min_j
  per (i = 0; i < n-1; i++) fer  // Cal fer n-1 iteracions
    min_j = i
    per (j = i + 1; i < n-1; i++) fer
      si taula[j] < taula[min_j] llavors
        min_j = j   // Actualitzem la posició si trobem un mínim
      fsi  // Hem trobat el més petit de la zona no ordenada
    fper
    intercanvia (taula[i], taula[min_j]) // Intercanviem posicions
    fper
```

Anàlisi del cost temporal de l'ordenació per selecció

Comentem ara com analitzar T(n), és a dir, el cost temporal en funció del nombre d'elements a ordenar. Com hem vist a teoria, cal tenir en compte el nombre d'iteracions en funció de n.

- En la 1a iteració del bucle extern, farà n - 1 iteracions
- En la 2a iteració, farà n - 2 iteracions
- En la 3a iteració, farà n - 3 iteracions...
- En la darrera iteració, el bucle intern farà només una iteració.

Tenim, doncs: T(n) = n - 1 + n - 2 + n - 3 + ... + 2 + 1, que es correspon amb una sèrie aritmètica. La suma dels seus elements es pot calcular com

$$T(n) = (n-1)\frac{(1+\ n-1)}{2} = \frac{n^2-n}{2}$$

L'ordre de T(n) és $O(n^2)$

Mesures de temps

Hi ha diferents formes de mesurar el temps d'execució d'un programa o funció en llenguatge C. En general, cal "consultar l'hora" abans i després d'executar el codi que volem "cronometrar" i fer una resta. Aquí presentem dues tècniques. Per utilitzar-les, cal incloure <time.h>

Mesurar segons

La funció ***time*( )**, que ja haureu utilitzat en la inicialització de la llavor per generar nombres aleatoris, ens retorna l'hora del sistema, però d'una manera especial: el nombre de segons transcorreguts des de l'1 de gener de 1970. Si restem els segons que passen entre dues mesures de temps (dues crides per prendre la mesura de temps) sabrem el nombre de segons que han transcorregut. El problema és si allò que mesurem triga menys d'un segon o si volem ser més precisos. Per tant, en aquesta pràctica, no obtem per aquest mètode.

Mesurar fraccions de segon

És més pertinent fer obtenir el temps en fraccions de segon. Una de les formes d'obtenir-ho és utilitzar els tics de rellotge que transcorren des que s'ha iniciat l'execució d'un programa. Això ho obtenim amb la funció ***clock*( )**, que retorna la informació en un tipus **clock_t**. Per calcular el nombre de segons com a nombre decimal, caldrà saber quants tics per segon fa el rellotge del sistema.

Per fer el càlcul hem de fer servir la constant **CLOCKS_PER_SEC**, definida també a <time.h>:

```
total_t = (double)(temps_f - temps_i) / CLOCKS_PER_SEC;
```

Tasques a fer

Al Campus Virtual teniu disponible un fitxer amb el codi font a completar. Fareu tot el desenvolupament en un únic fitxer. L'objectiu de les tasques és prendre mesures de temps i comptar el nombre d'iteracions que fa el programa en funció del nombre d'elements que conté la taula. El codi proporcionat ja conté algunes funcions, examineu-lo.

Tasca 1. Completeu la funció *ordenar*( ) d'acord amb l'algorisme que us hem proporcionat en aquest enunciat. Comproveu que l'ordenació implementada us funciona.

Tasca 2. Afegiu el codi per mesurar el temps de generació de les dades i per mesurar el temps d'ordenació. Caldrà invocar quatre vegades la funció *clock*( ). Veureu que ja disposeu de quatre variables definides, que haureu d'utilitzar.

Tasca 3. Veureu que la funció *ordenar*( ) retorna un enter: és el nombre d'iteracions que es fan per ordenar les dades. Aquest valor l'hauríeu de definir com un "long", ja que prendrà valors elevats, i anar-lo incrementant dins el bucle intern de la funció (el que té "j" com a variable de control).

Tasca 4. Finalment, fareu una sèrie d'experiments, cadascun d'ells amb un nombre d'elements diferent. Per exemple, podeu començar amb 100 i arribar desenes de milers. Per a cada execució, el programa us mostrarà per pantalla el temps esmerçat i el nombre d'iteracions. Cal dirigir els experiments a respondre les preguntes següents:

- En funció del nombre d'elements de la taula d'enters, què consumeix més temps: emplenar la taula o ordenar la taula? Quin és el motiu?
- Quantes iteracions fa *ordenar*( ) en funció dels diferents valors per a nombre d'elements que heu testejat?

Tasca 5 (opcional). Si us queda temps, modifiqueu la funció que emplena la taula de dades de manera que ara no les empleni amb valors aleatoris sinó creixents (per exemple, a cada posició hi assigneu el valor de l'índex del bucle). Repetiu els experiments i valoreu com es comporta l'ordenació per selecció quan les dades ja estan ordenades.

Lliurament

Per tal que aquesta pràctica compti per l'avaluació heu d'adjuntar els següents fitxers a la tasca corresponent, abans de la data que s'hi indica:

1. Un PDF amb els fitxer de codi font (degudament comentat i indentat), el resultat de les proves que heu fet, i les respostes a les preguntes de la tasca 4.
2. El codi font.

El professor es reserva el dret d'avaluar els vostres lliuraments. Si aquests s'han fet de forma fraudulenta (còpia, document en blanc, o que no té res a veure amb la solució) s'aplicarà el que especifica la Guia de l'assignatura.